

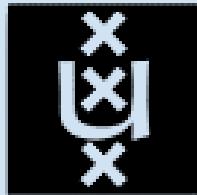
Conversational Recommendation: Formulation, Methods, and Evaluation

Wenqiang Lei, Xiangnan He, Maarten de Rijke, Tat-Seng Chua

wenqanglei@gmail.com, hexn@ustc.edu.cn, derijke@uva.nl, dcscts@nus.edu.sg

slides will be available at: <https://core-tutorial.github.io>

A literature survey based on this tutorial as well as other materials will be available soon.



UNIVERSITY
OF AMSTERDAM

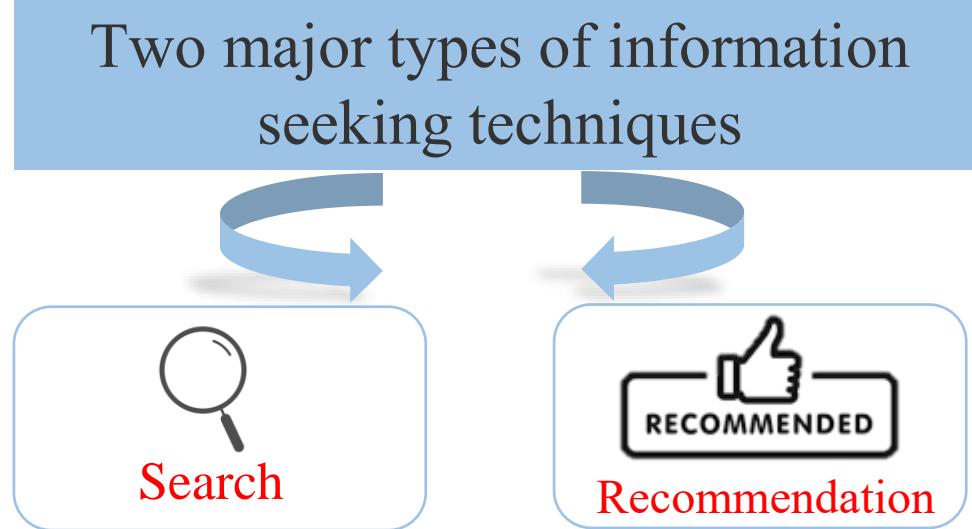


Ahold
Delhaize

• Information Seeking

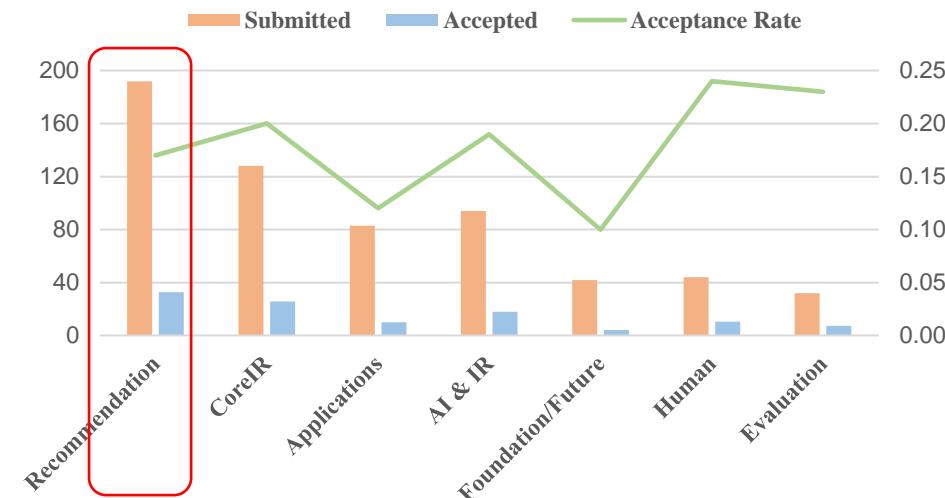
Information explosion problem?

- Information seeking requirements
 - E-commerce(Amazon and Alibaba)
 - Social networking(Facebook and Wechat)
 - Content sharing platforms(Instagram and Pinterest)

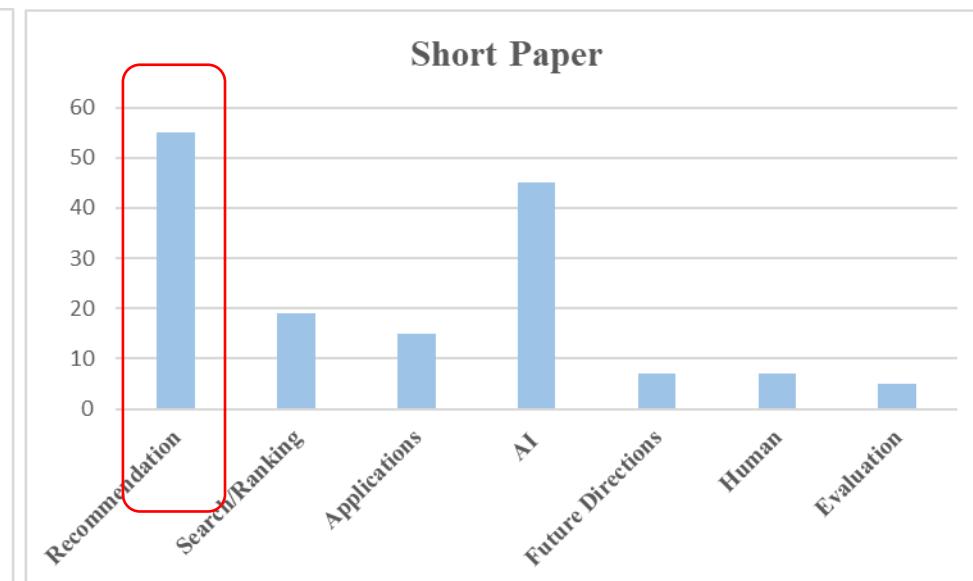
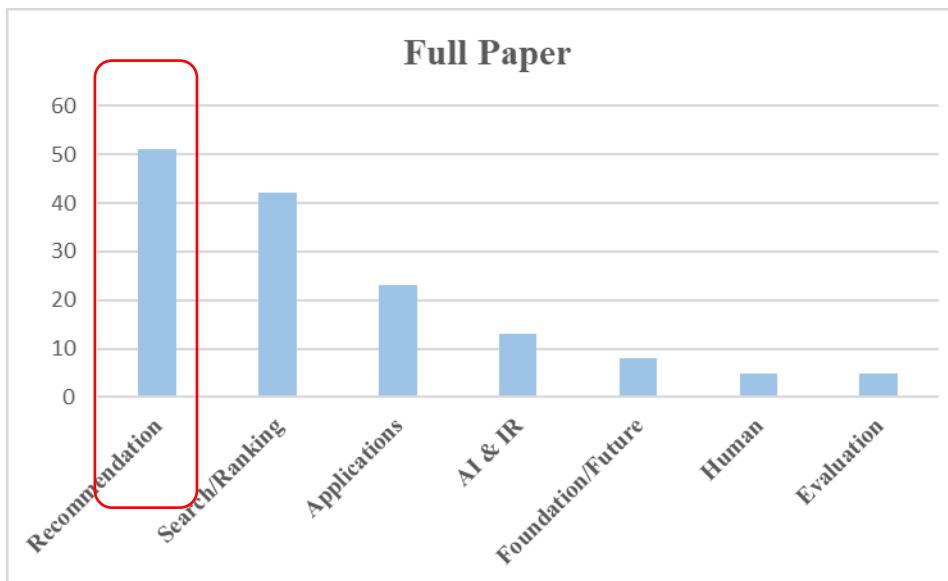


• Recommendation Has Become Prevalent in IR Community

Recommendation becomes the most popular track



2019 SIGIR Hot Topics



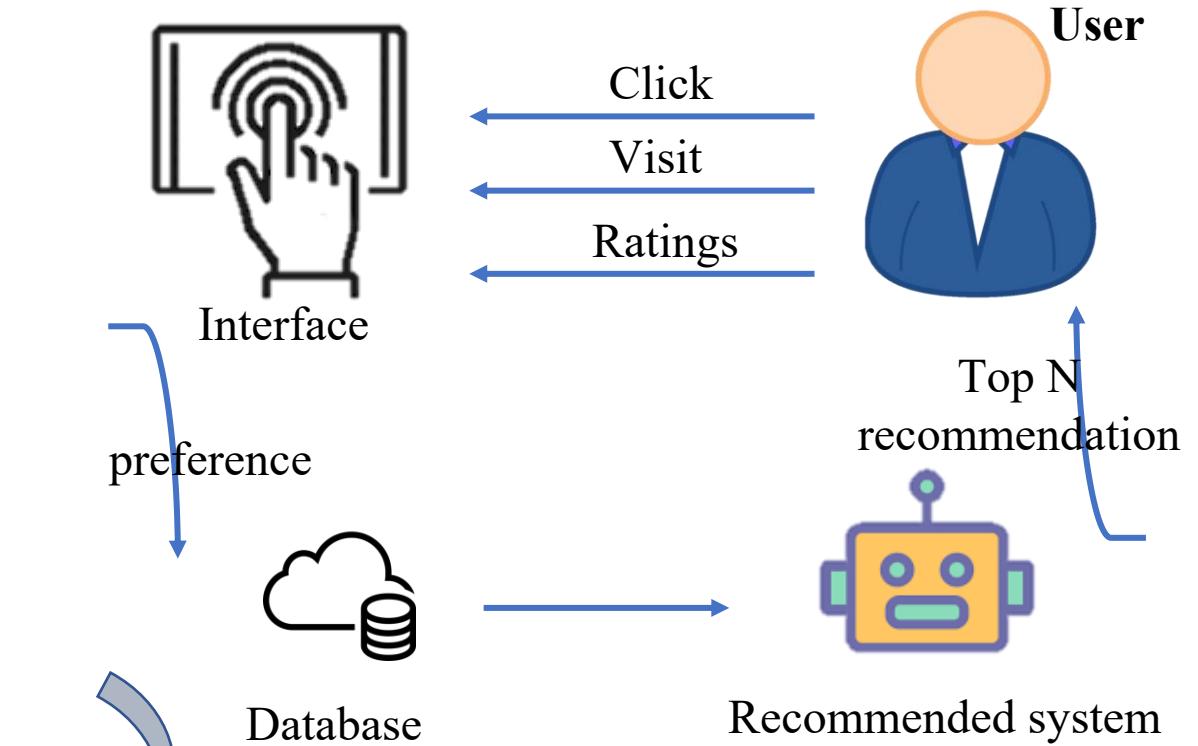
SIGIR of different Topics were received in 2020

• Typical Recommender Systems

Recommender systems

- predict a user's **preference** towards an item by analyzing their **past behavior**
(e.g., click history, visit log, ratings on items, etc)

Implicit



• Existing Static Recommendation: Collaborative Filtering

➤ Collaborative filtering

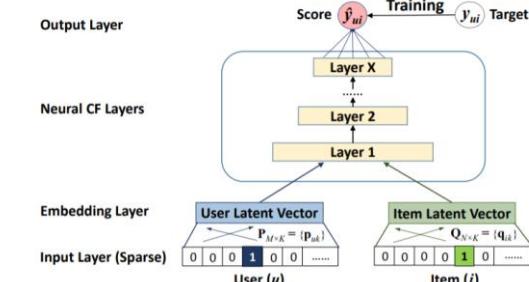
- matrix factorization and factorization machines

Feature vector x										Target y											
$x^{(1)}$	1	0	0	...	1	0	0	0	...	5 $y^{(1)}$											
$x^{(2)}$	1	0	0	...	0	1	0	0	...	3 $y^{(2)}$											
$x^{(3)}$	1	0	0	...	0	0	1	0	...	1 $y^{(3)}$											
$x^{(4)}$	0	1	0	...	0	0	1	0	...	4 $y^{(4)}$											
$x^{(5)}$	0	1	0	...	0	0	0	1	...	5 $y^{(5)}$											
$x^{(6)}$	0	0	1	...	1	0	0	0	...	1 $y^{(6)}$											
$x^{(7)}$	0	0	1	...	0	0	1	0	...	5 $y^{(6)}$											
A	B	C	...	T1	NH	SW	ST	...	T1	NH	SW	ST	...	Other Movies rated	Time	T1	NH	SW	ST	...	Last Movie rated
User	Movie	...																			

Factorization Machines

➤ Deep learning approaches

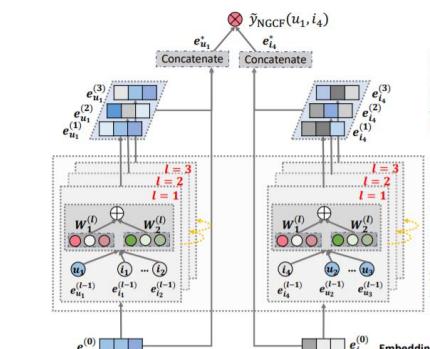
- neural factorization machines & deep interest networks



Neural Collaborative Filtering

➤ Graph-based approaches

- expressiveness and explainability of graphs

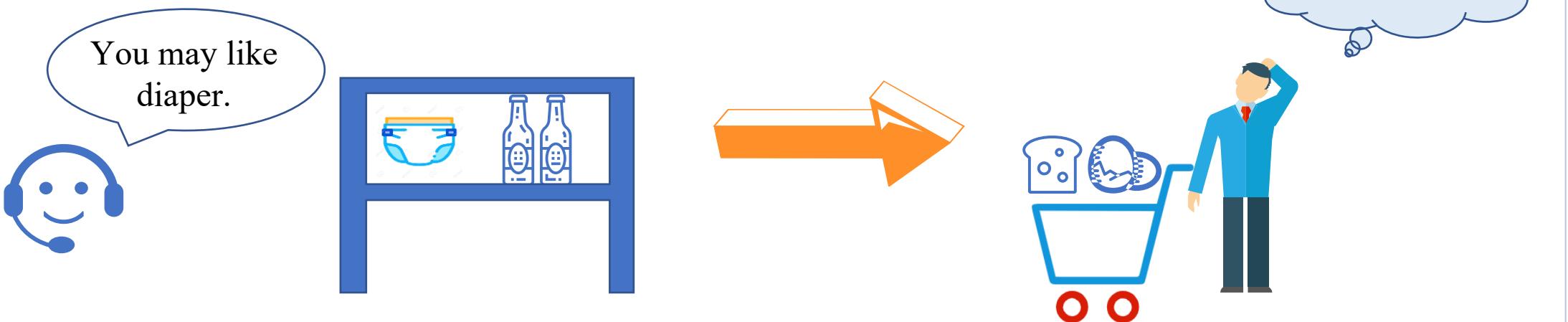


Neural Graph Collaborative Filtering

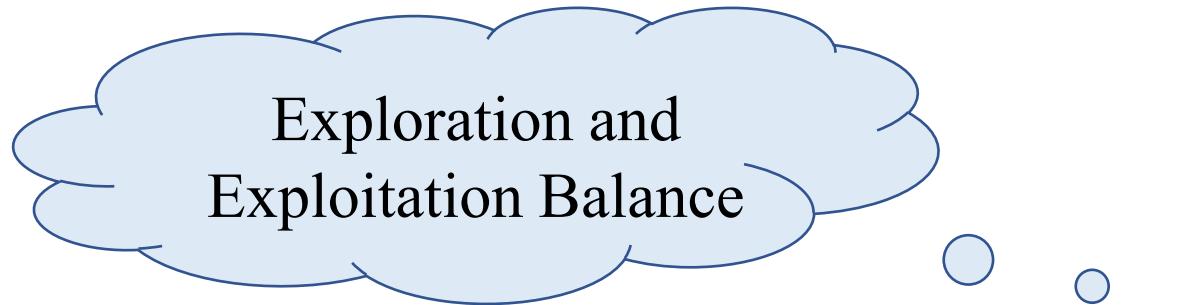
• Limitation: Information Asymmetry

Key Problems for Recommendation: Information Asymmetry

- Information asymmetry
 - A system can only **estimate** users' preferences based on their historical data
- Intrinsic limitation
 - Users' preferences often **drift** over times.
 - It is hard to find **accurate reasons** to recommendation



• Existing Online Recommendation: Bandit



Online Recommendation:

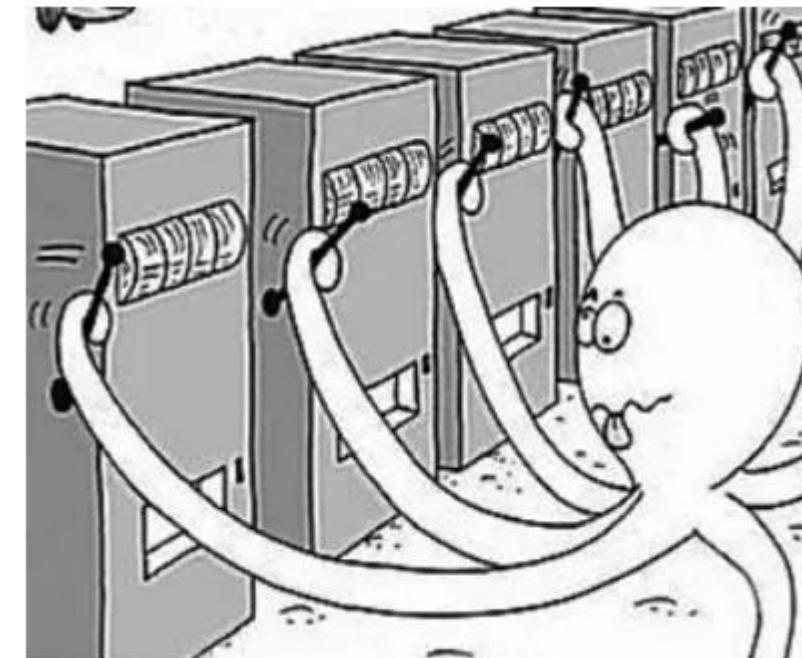
Arm → Item/ Item Category

Reward → User feedback

Environment → User

□ Bandit Algorithm:

- Exploit-Explore problem
- Cold-Start problem



Multi-Armed Bandit

• Limitation: Lack of Explainability

A model still has no channel to know find the exact reason why a user prefer an item.

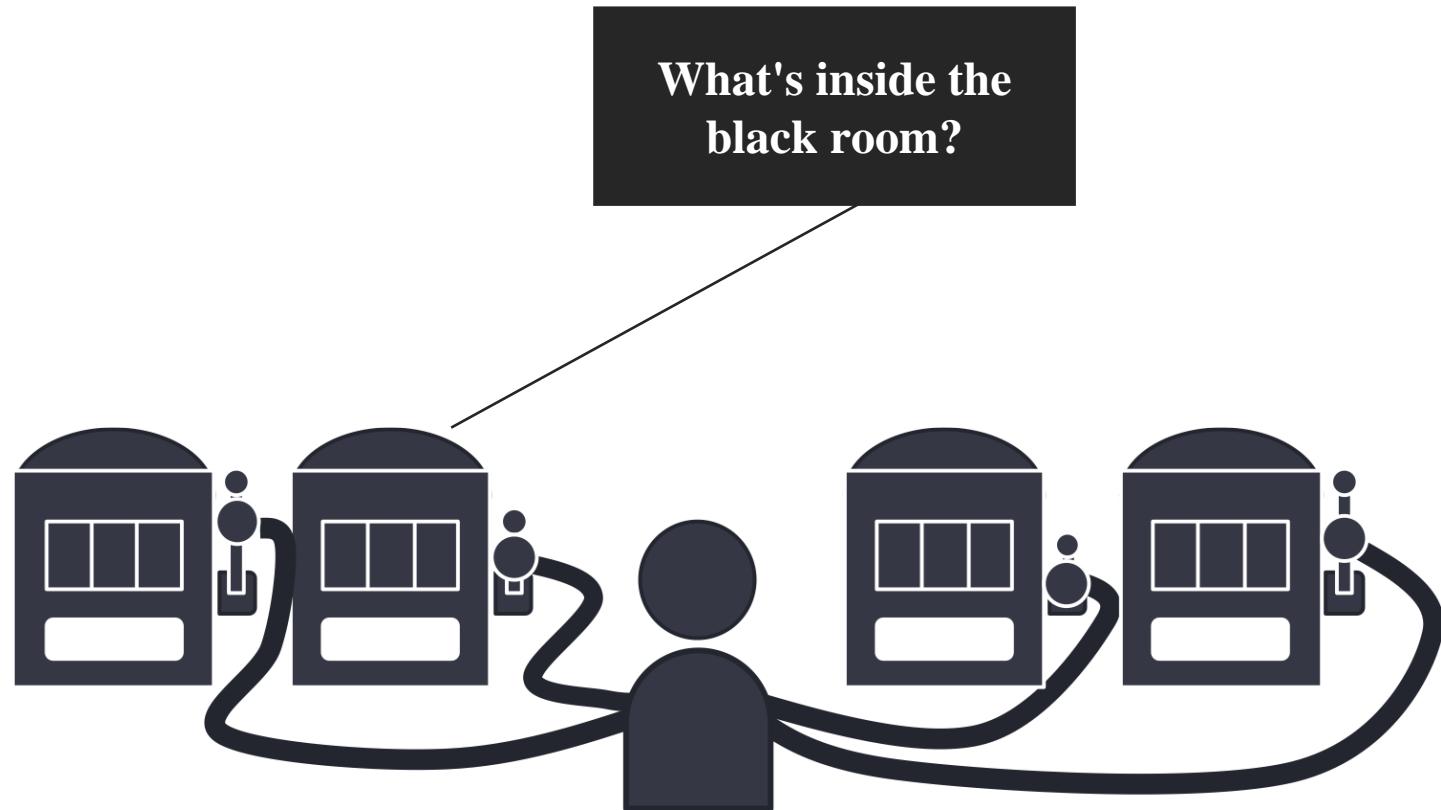


Figure credit: [Spotx](#)

• Conversation Brings Revolution

A screenshot of a conversational recommender system interface. The user asks for a mobile phone on Amazon, specifying an Android operating system, a screen size larger than 5 inches, at least 64 Gigabytes of storage, and no preference for phone color. The bot suggests five phones from Samsung and Huawei, all within 700 dollars. The user selects the first phone and the bot places the order.

Can you find me a **mobile phone** on Amazon?
Sure, what **operating system** do you prefer?

I want an **Android** one.
OK, and any preference on **screen size**?

Better larger than **5 inches**.
Do you have requirements on **storage capacity**?

I want it to be at least **64 Gigabytes**.
And any preference on **phone color**?

Not particularly.
Sure, then what about the following choices?



I don't like them very much...
OK, do you have any preference on the **brand**?

Better be **Samsung or Huawei**.
Any requirement on **price**?

Should be **within 700 dollars**.
OK, then what about these ones?



Great, I want the first one, can you order it for me?
Sure, I have placed the order for you, enjoy!

A screenshot of a conversational recommender system interface. The user is looking for a dance music artist. The bot asks if the user likes rock and pop music. The user confirms liking both. The bot then recommends Michael Jackson as a music artist the user might like.

Hi! I'm looking for a **dance** music artist.

Do you like **rock** music?

Yes! I like it!

Do you like **pop** music?

Yes! I like it!

You may like music artist
Michael Jackson!

Yes! Thank you!

Conversational Recommender Systems

➤ Interactive recommendation

➤ Using natural languages

The example of a conversational recommender system

- Conversational Recommender Systems In a Broader Perspective
- Tag-based Interaction

Skip

Choose your interests

Get personalized video recommendations

Animals Comedy

Travel Food

Sports Beauty & Style

Art Gaming

Science & Education

Dance DIY

Next

The example of tag-based interaction on tiktok

Dislike the content? Choose the ones that interest you

Humor Music Celebrity

Animation Affection Drama

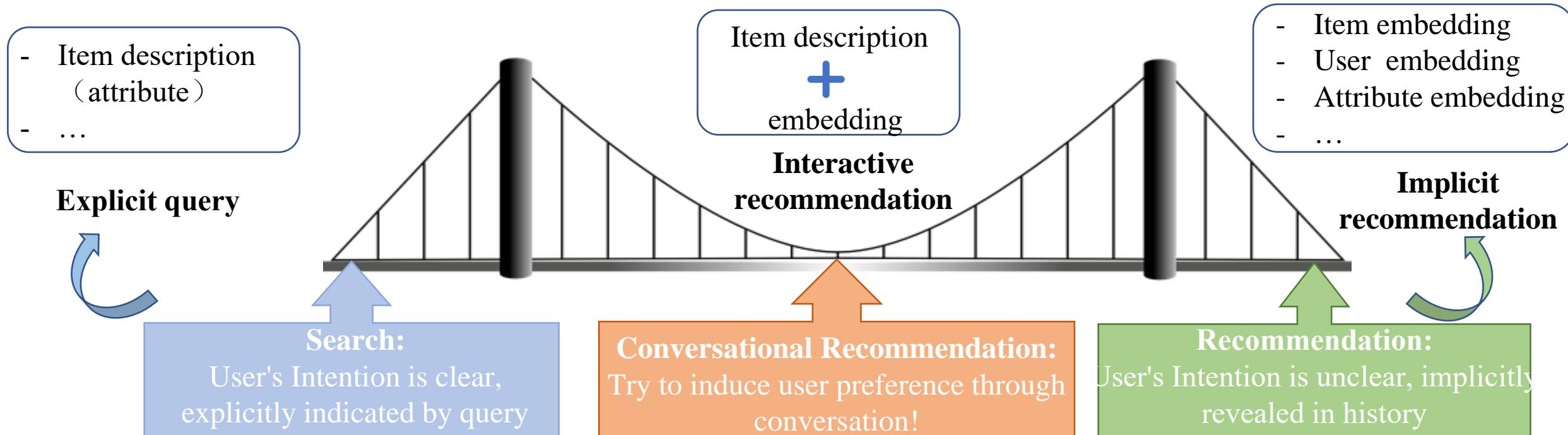
Science Fitness Pets

Select at least 4 interests

The example of tag-based interaction on kuaishou

• Conversational Recommendation Bridges Search and Recommendation

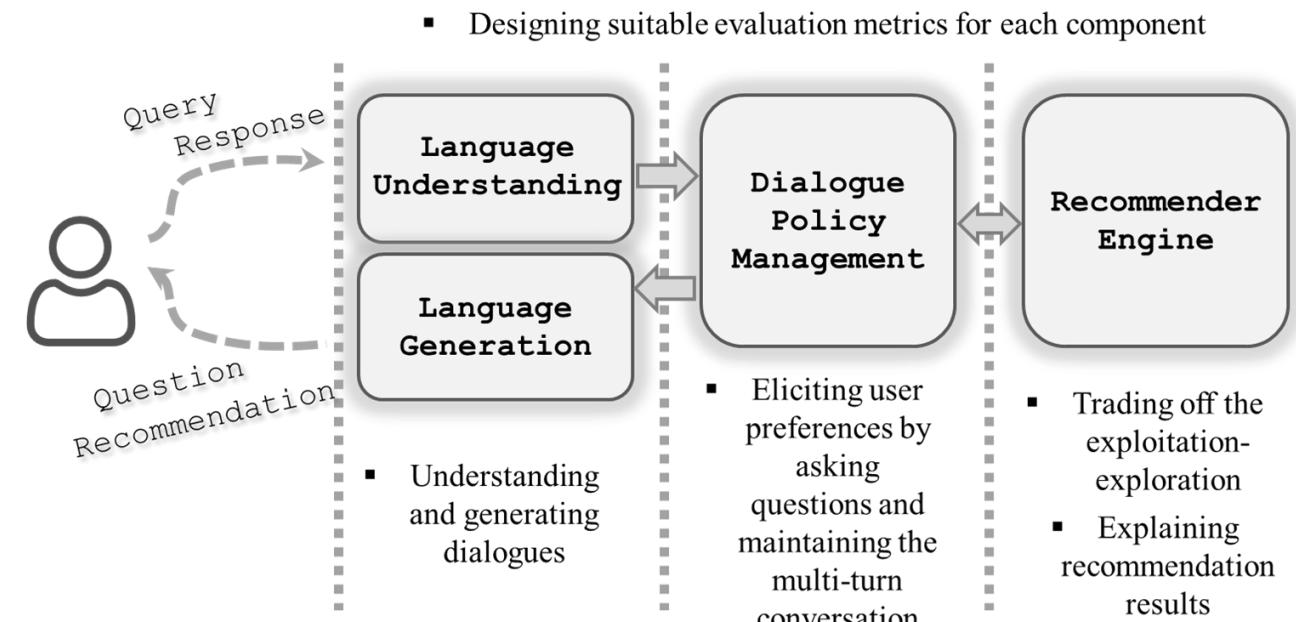
Traditional paradigms for information-seeking:
Search (pull) or Recommendation (push)



• Conversational Recommender Systems

Four Directions being Explored

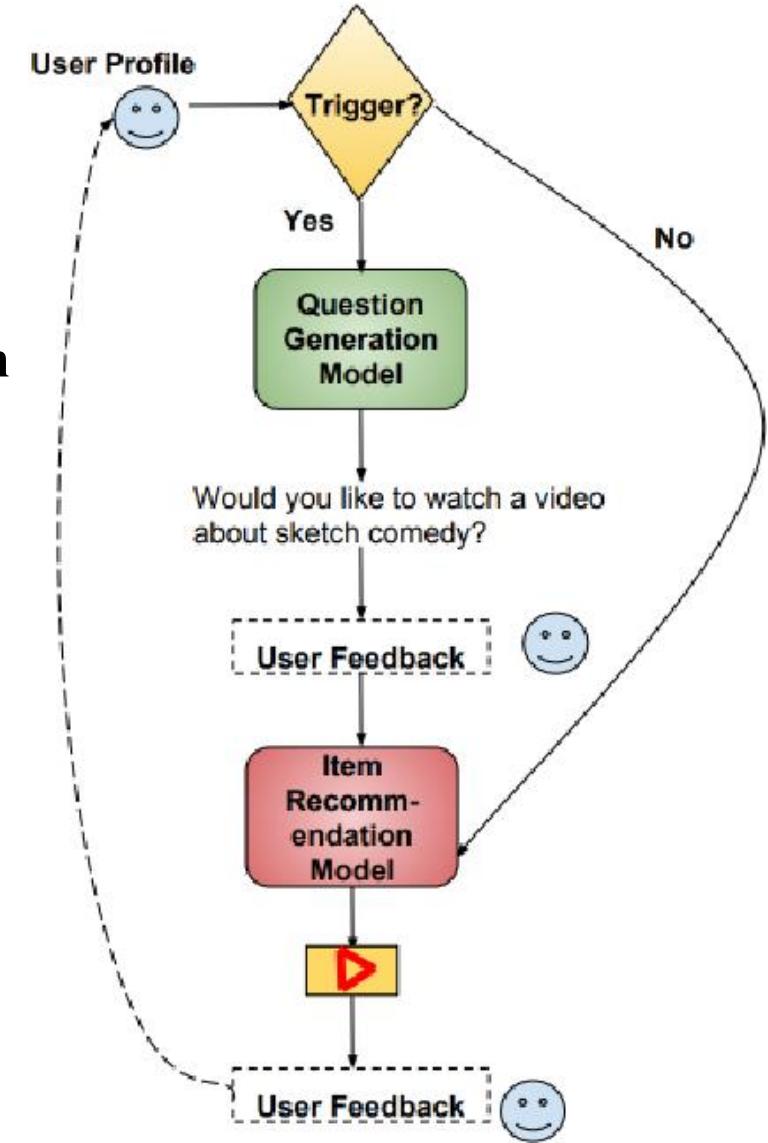
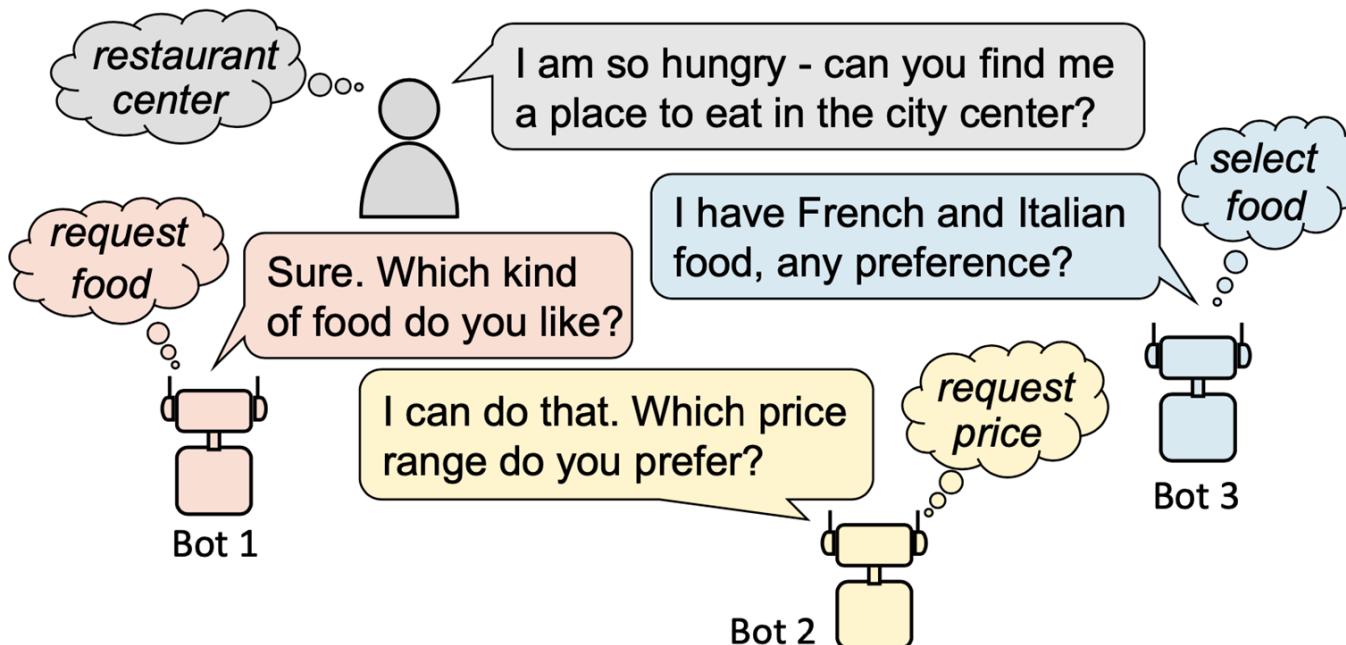
1. Question Driven Approaches
2. Multi-turn Conversational Recommendation Strategy
3. Exploitation-Exploration Trade-offs for Cold Users
4. Dialogue Understanding and Generation



• Question Driven Approaches in CRS

The key advantage of conversational recommendation:
being able to ask questions.

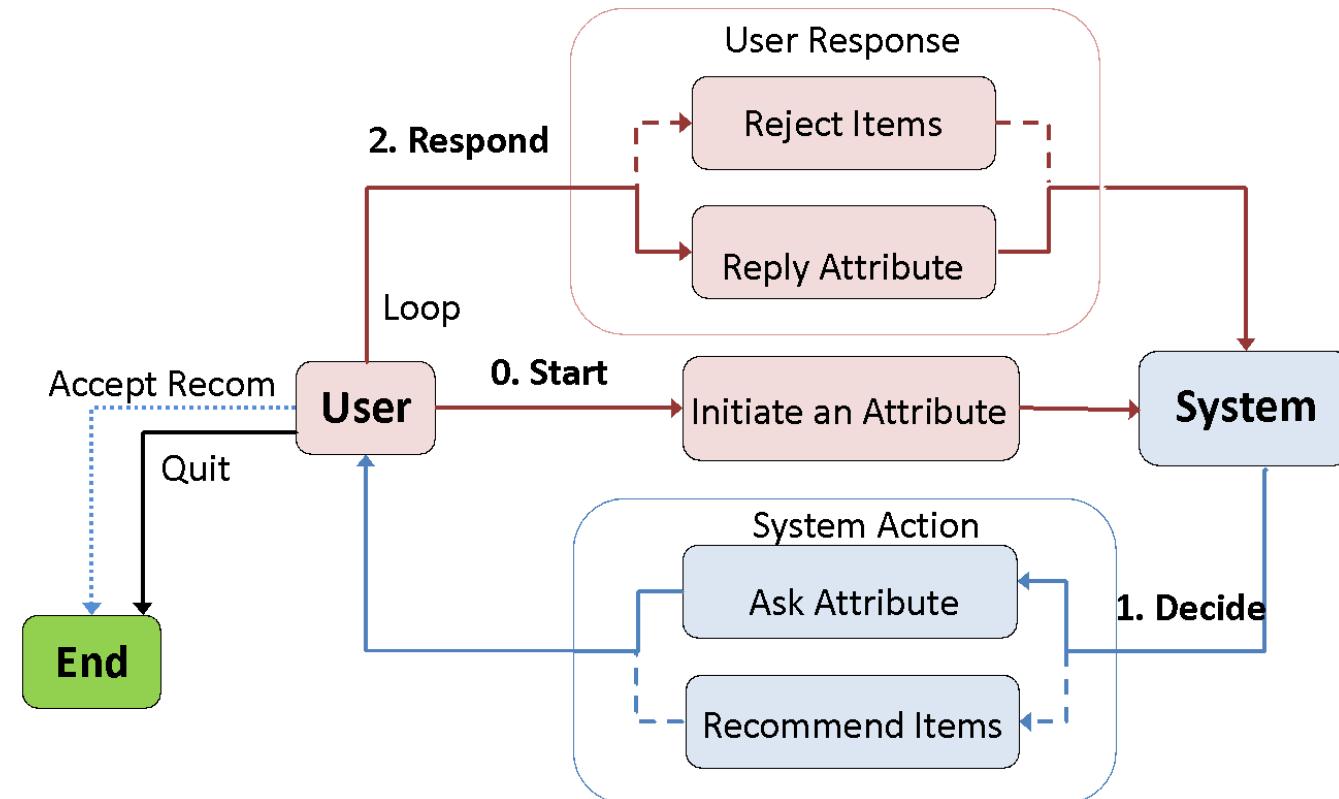
- Ask about **attributes/topics/categories** of items to narrow down the recommended candidates.



•Multi-turn Conversational Recommendation Strategy

A System needs to choose to **ask questions** and **make recommendations** in a multi-turn conversation

- **Purpose:** making successful recommendations with less turns of interactions
- **Challenges to address:**
 1. Which items or attributes to recommend?
 2. When to ask questions and when to make recommendations?
 3. How to adapt user feedback



Lei et al. “Estimation–Action–Reflection: Towards Deep Interaction Between Conversational and Recommender Systems” (WSDM’20)

• Exploitation-Exploration Trade-offs for Cold Users

Exploration
(Learning)

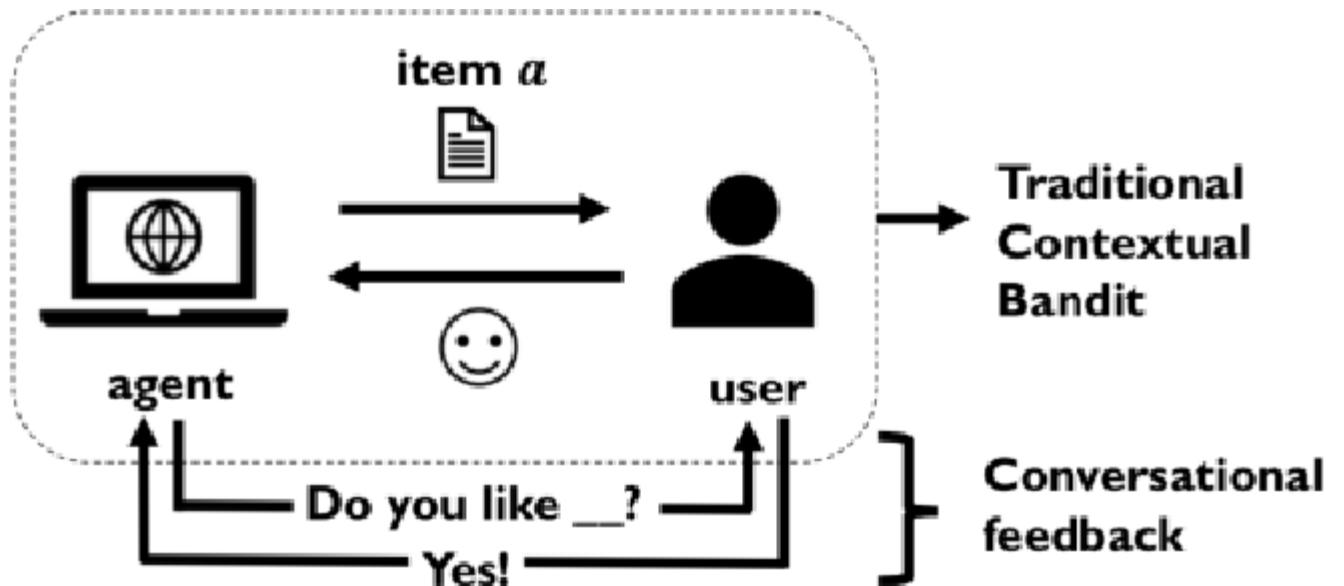
Trade-off

Take some risk to collect information
about unknown options

Exploitation
(Earning)

Takes advantage of the best
option that is known.

- ✓ Leverage the dynamics of CRS to benefit the E&E trade-off for cold users/items.



• Dialogue Understanding and Generation



Rule/Template-based

Inflexible,
constrained

Fail to understand
user intent.

I want some
music.

What category of music do you like?

Pop.

Which Pop singer do you like?

Jay Chou.

Hope you enjoy this song:



七里香 Qi-Li-Xiang
By Jay Chou

Change it.

Hope you enjoy this song:



Change it
By Stevie Ray
Vaughan

Neural methods



I want some
music.

Feel tired in work? What do you want?

Yeah, wanna some relaxed music

As you wish, how about this one?

It is a new song just released by Jay Chou



Mojito
By Jay Chou

Oh, I love it! But I have listened it like 100
times. I wanna try something new.

Yeah, Mojito is too popular these day.
Maybe you like some niche songs like
this one. The singer is also Jay Chou.



麦芽糖 Malt Candy
By Jay Chou

Casual, more
natural.

Extract intent from
user utterances.

Express actions in
generated responses

Fluent and
Consistent.

• Tutorial Outline

❑ A Glimpse of Dialogue System

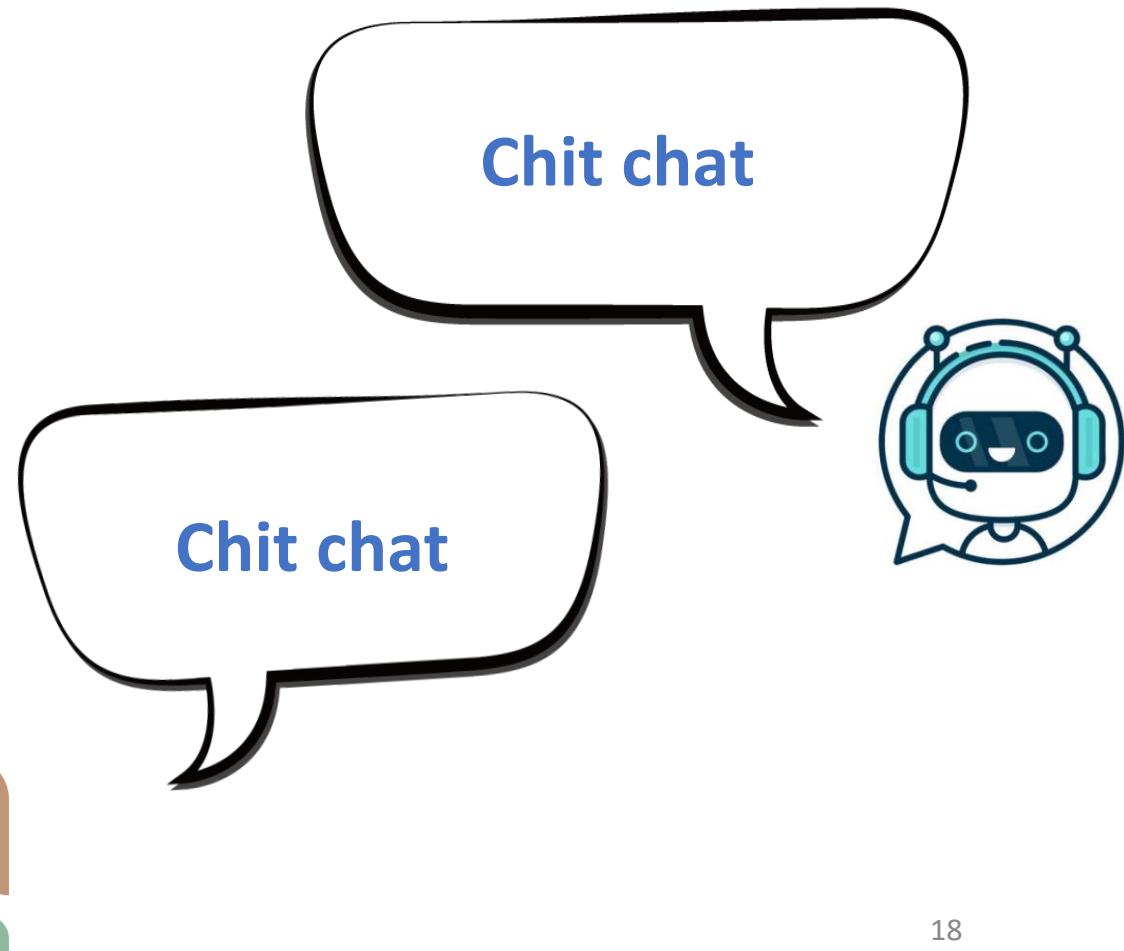
- ❑ Four research directions in conversational recommendation system
 - ❑ Question Driven Approaches
 - ❑ Multi-turn Conversational Recommendation Strategy
 - ❑ Dialogue Understanding and Generation
 - ❑ Exploitation-Exploration Trade-offs for Cold Users
 - ❑ Summary of Formalizations and Evaluations

• Two Types of Dialogue Systems

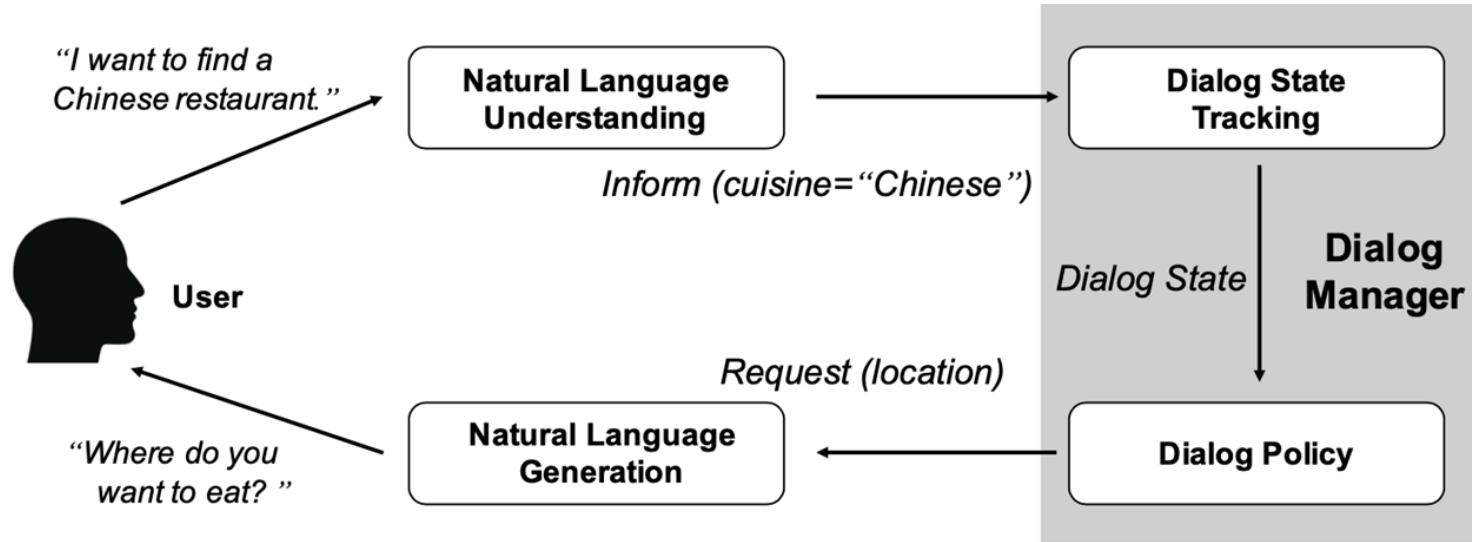
- Task-oriented Dialogue System



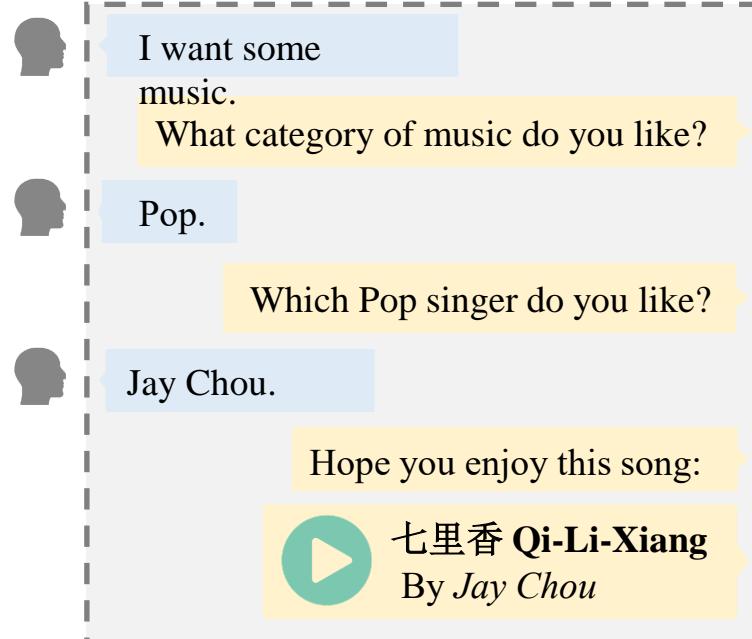
- Non-task-oriented Dialogue System (Chatbot)



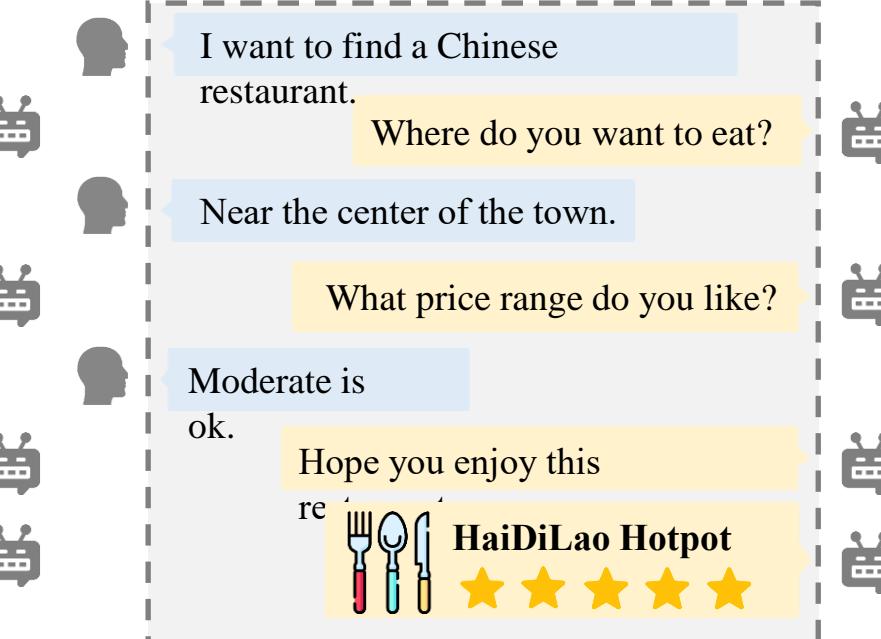
• Typical Structure of Task-oriented Dialogue System



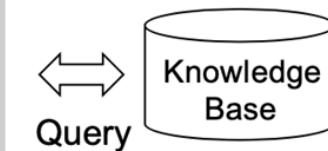
Recommending music



Booking restaurants

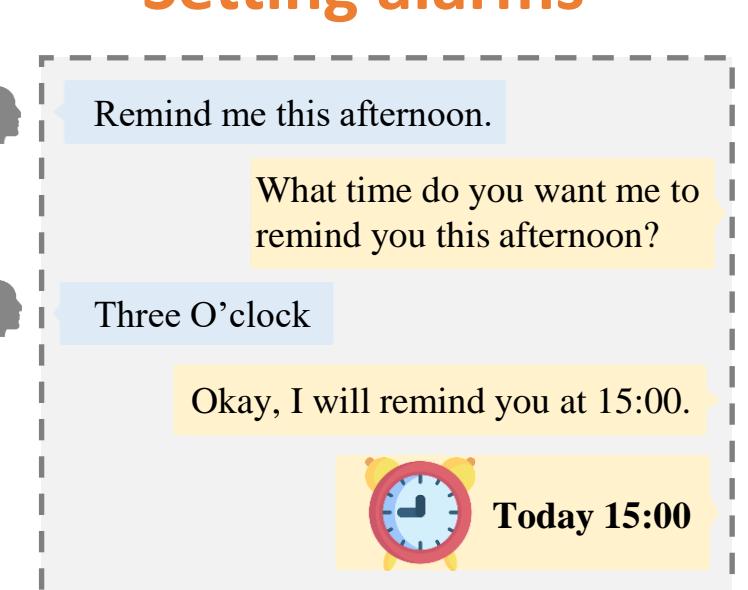


Classical pipeline structure



Zhang et al. Recent advances and challenges in task-oriented dialog system (Science China '20)

Setting alarms



• Natural Language Understanding

• Three Purpose:

1. Domain detection

2. Intent detection

3. Slot value extraction

An example utterance with annotations in IOB format

W	find	recent	comedies	by	james	cameron
S	↓	↓	↓	↓	↓	↓
O	O	B-date	B-genre	O	B-dir	I-dir
D	movies					
I	find_movie					

where:

S: semantic slots.

D: domain.

I: intent.

In IOB format:

O: a token belongs to no chunk.

B-: the beginning of every chunk.

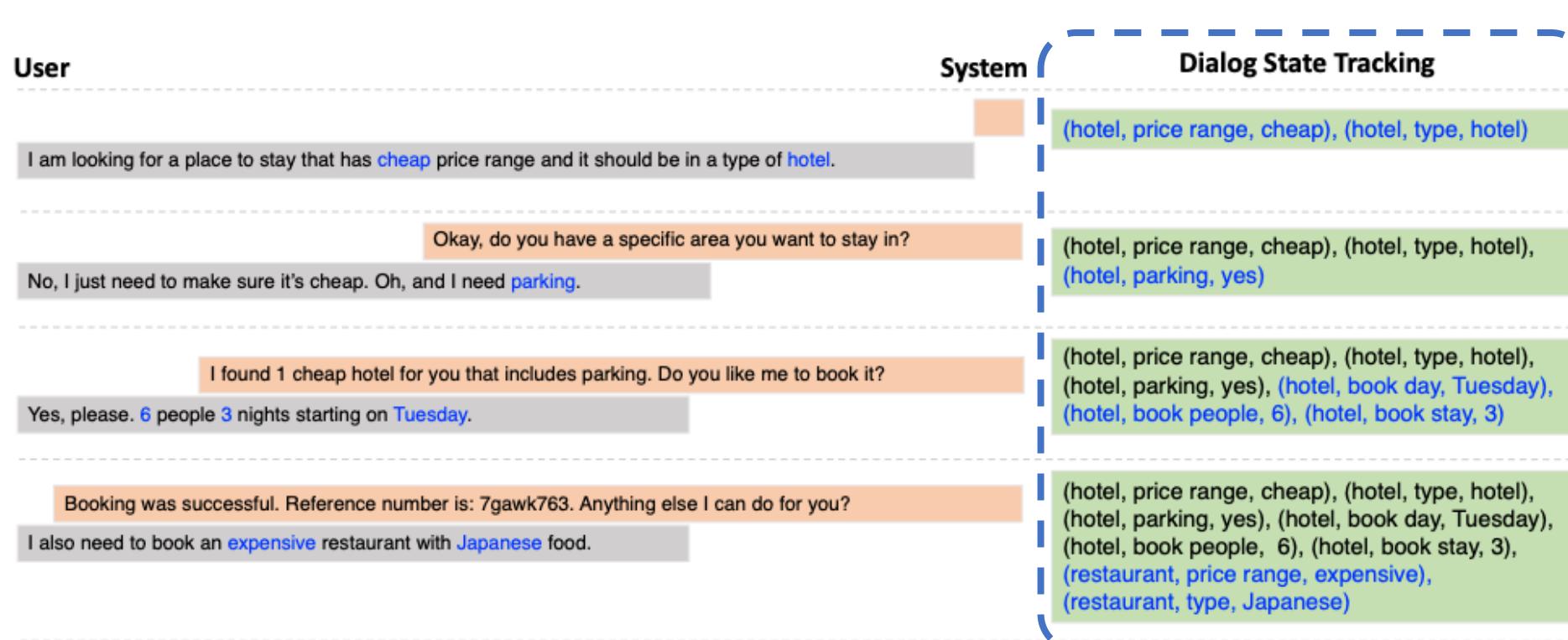
I-: a token inside a chunk

• Dialogue State Tracking

Aiming to track all the states
accumulated across the conversational
turns

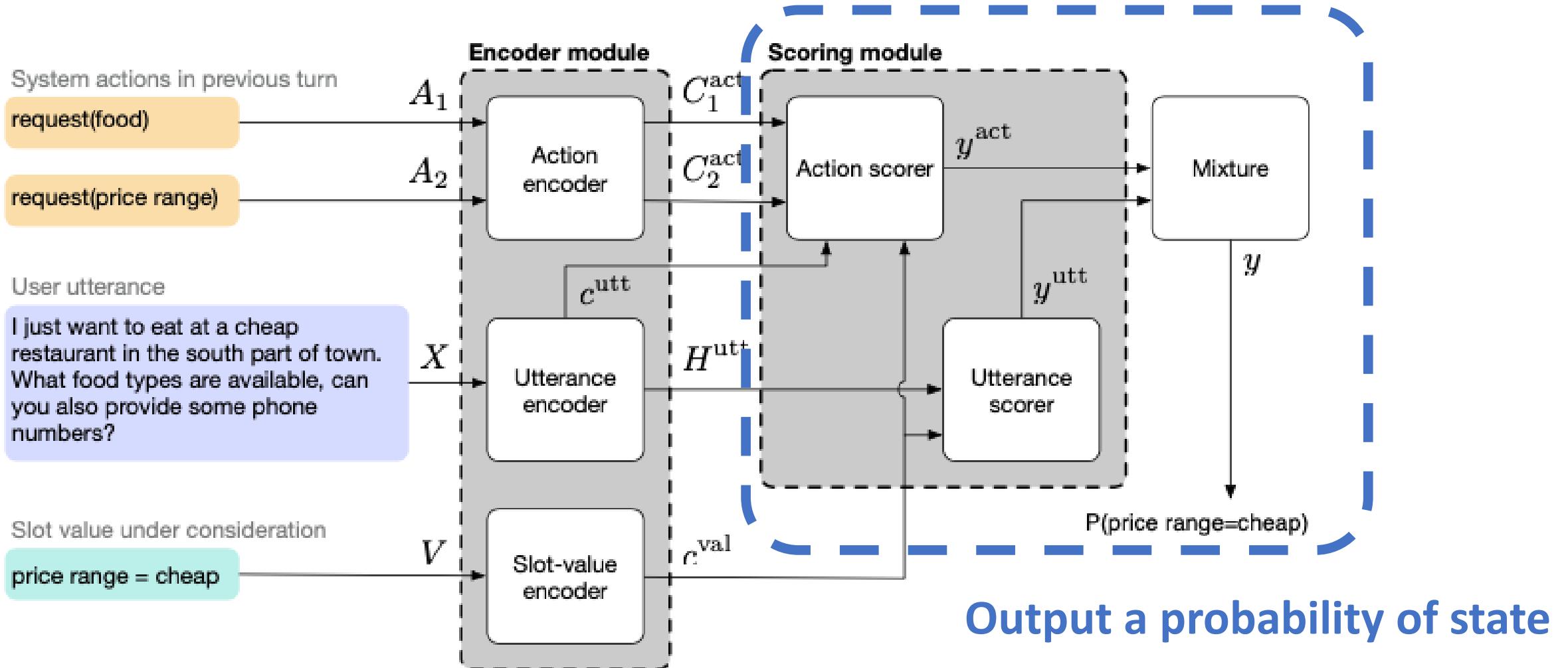
Recent solutions: latent vector-based
methods

1. Classification (picklist-based).
2. Copying (generative)



- Jointly Solving Natural Language Understanding and Dialogue State Tracking -- Classification

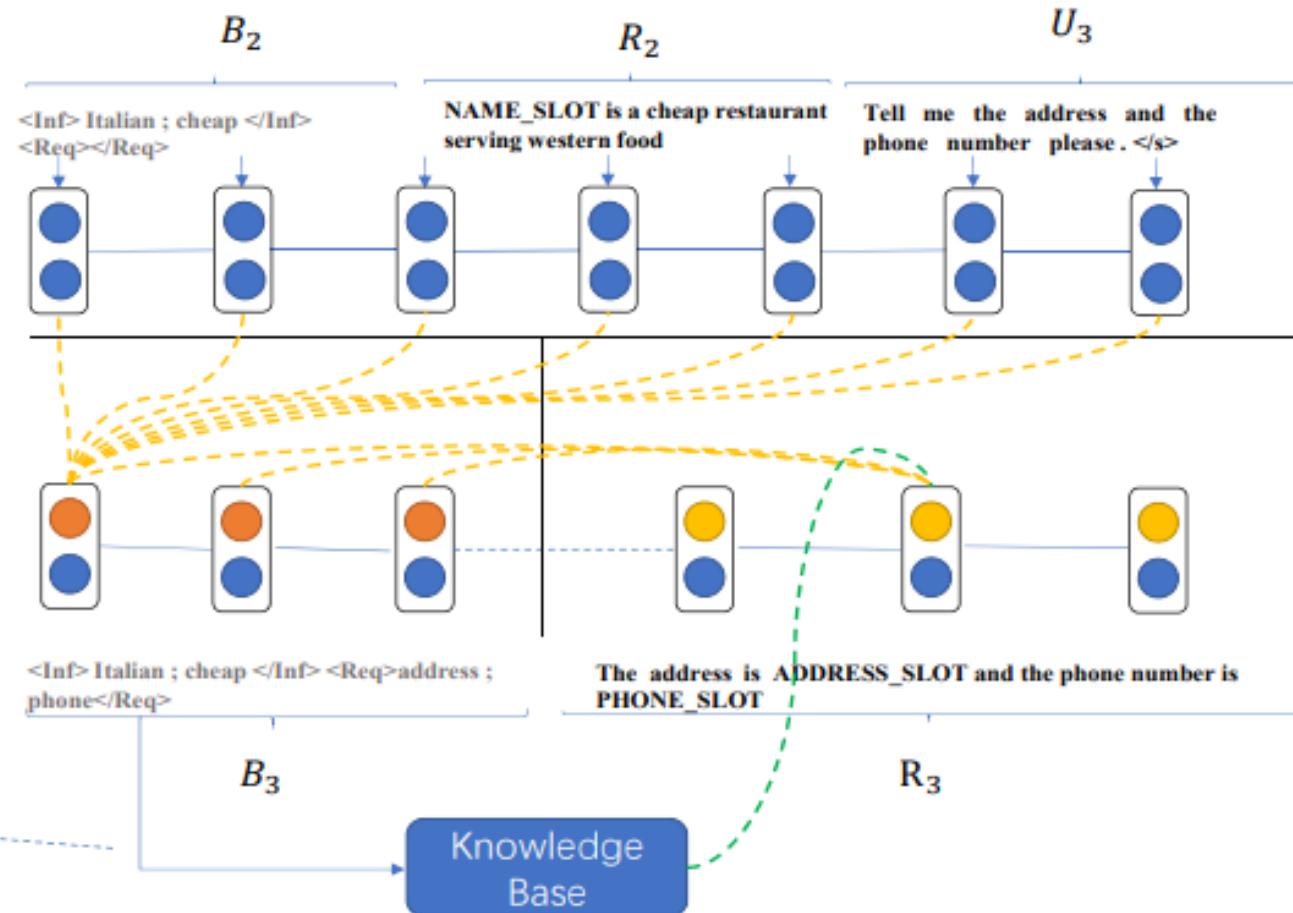
- Using a classifier as dialogue state tracker



- Jointly Solving Natural Language Understanding and Dialogue State Tracking -- Copying

- Find the text span in original utterances.

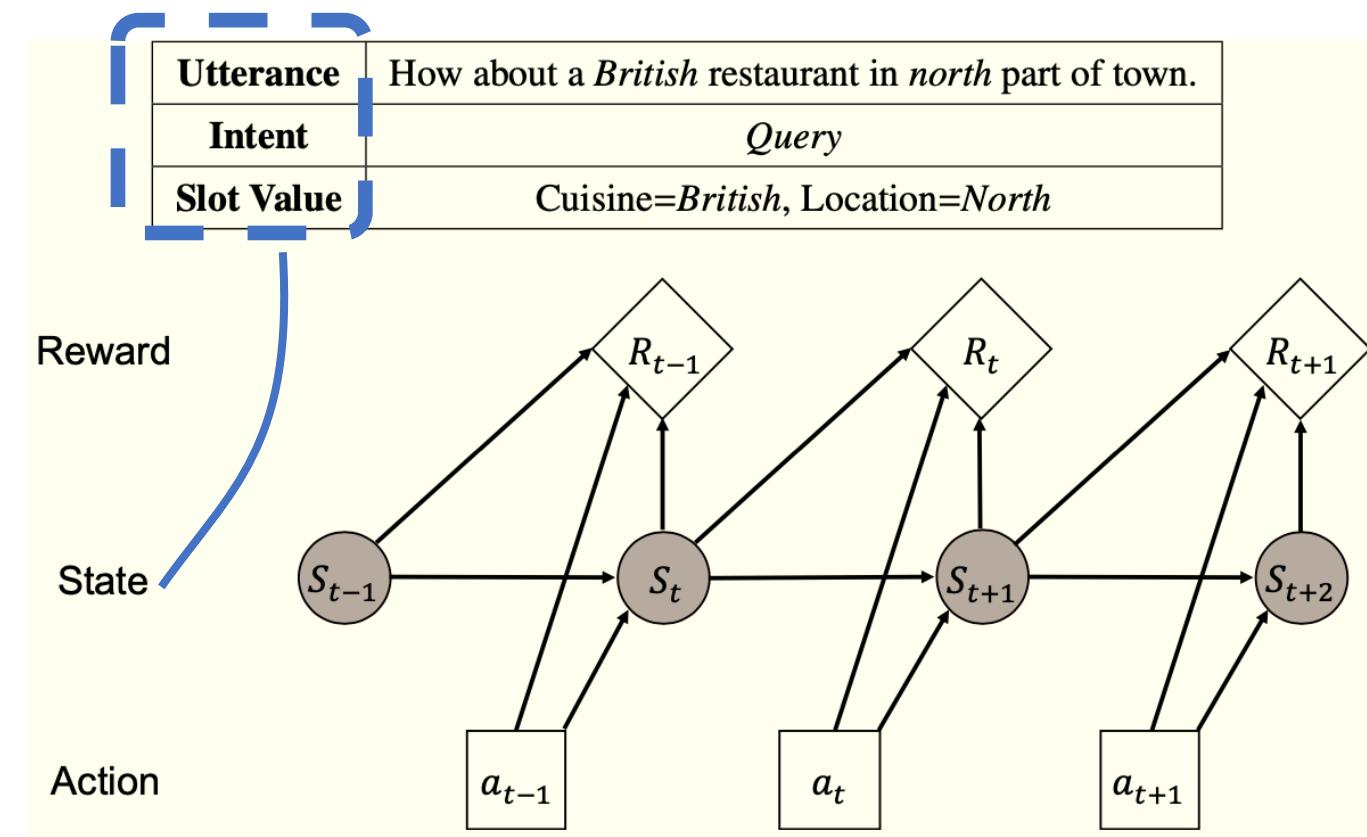
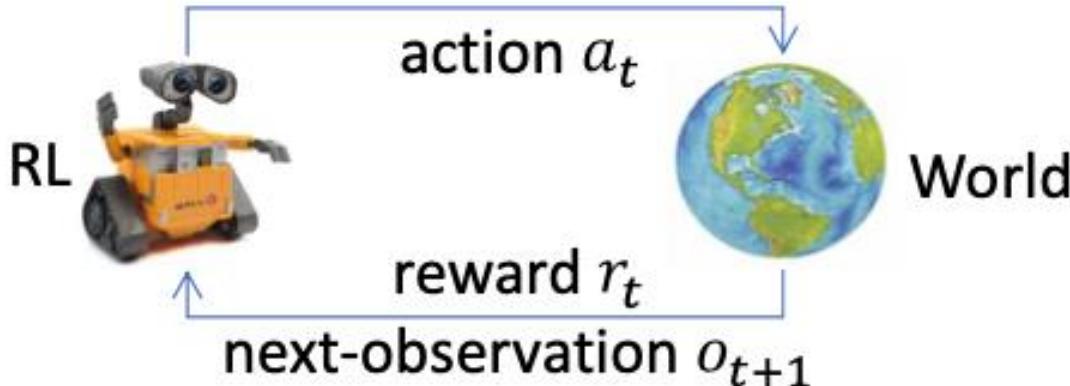
Turn	Dialogue
User ₁	Can I have some Italian food please?
Machine ₁	<Inf> Italian </Inf><Req> </Req> What price range are you looking for?
User ₂	I want cheap ones.
Machine ₂	<Inf> Italian ; cheap </Inf> <Req></Req> NAME_SLOT is a cheap restaurant serving western food
User ₃	Tell me the address and the phone number please .
Machine ₃	<Inf> Italian ; cheap </Inf> <Req>address ; phone</Req> The address is ADDRESS_SLOT and the phone number is PHONE_SLOT



• Dialogue Policy

- Dialogue act in a session are generated sequentially, so it is formulated as a **Markov Decision Process (MDP)**

- Can be address by **Supervised Learning or Reinforcement Learning**



A framework of MDP.

Zhang et al. Recent advances and challenges in task-oriented dialog system (Sci China Tech Sci' 20)

• Natural Language Generation

• Strategies:

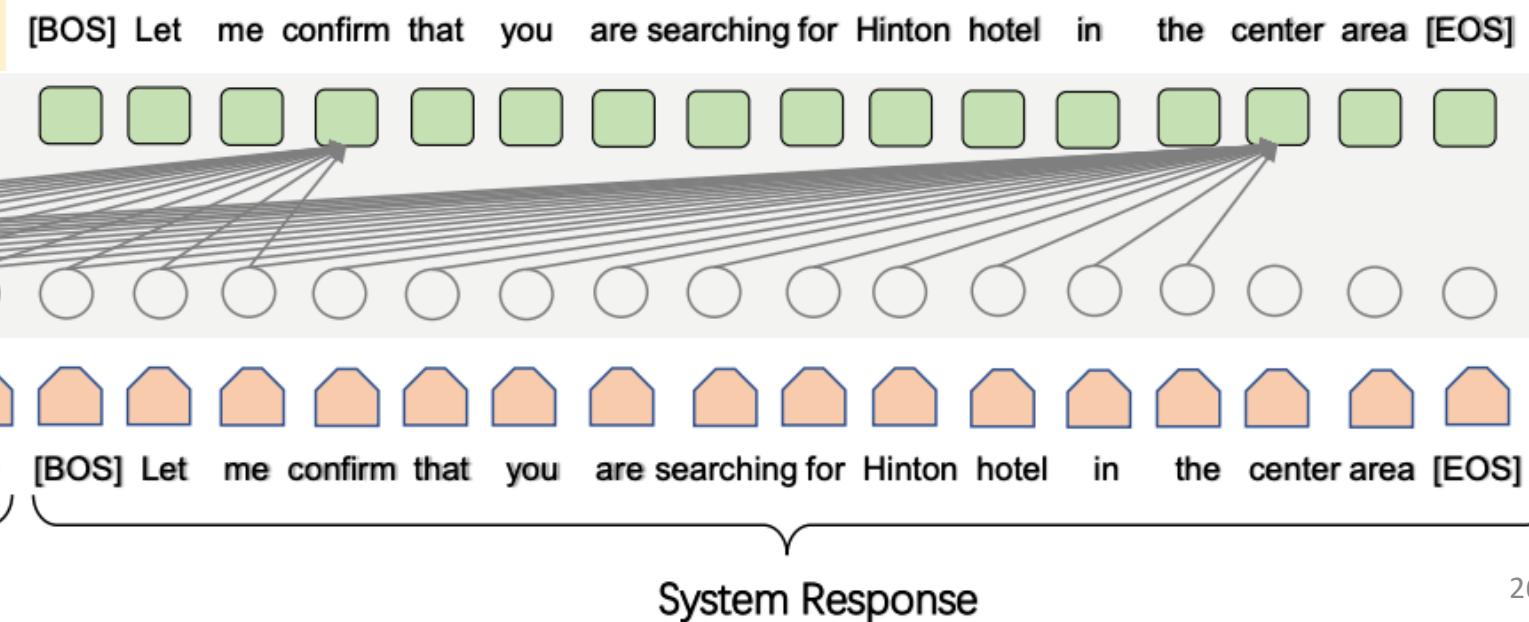
- Surface realization
- Conditioned language generation
(RNN-based neural network)

• Challenges:

- **Adequacy**: meaning equivalence,
- **Fluency**: syntactic correctness,
- **Readability**: efficacy in context,
- **Variation**: different expression.

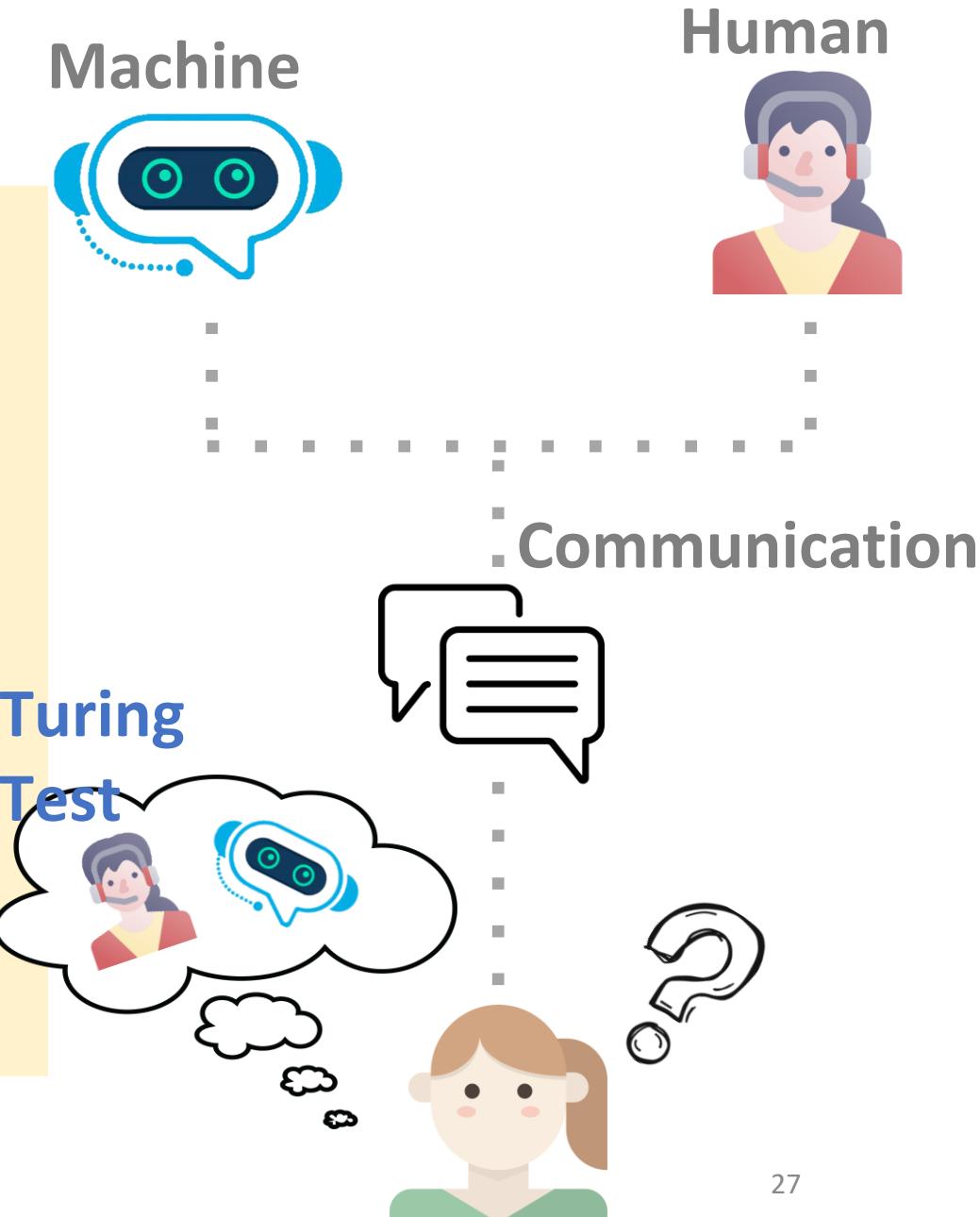
Semantically-Conditioned Generative Pre-Training (SC-GPT) Model

Peng et al. Few-shot Natural Language Generation for Task-Oriented Dialog (Arxiv' 20)

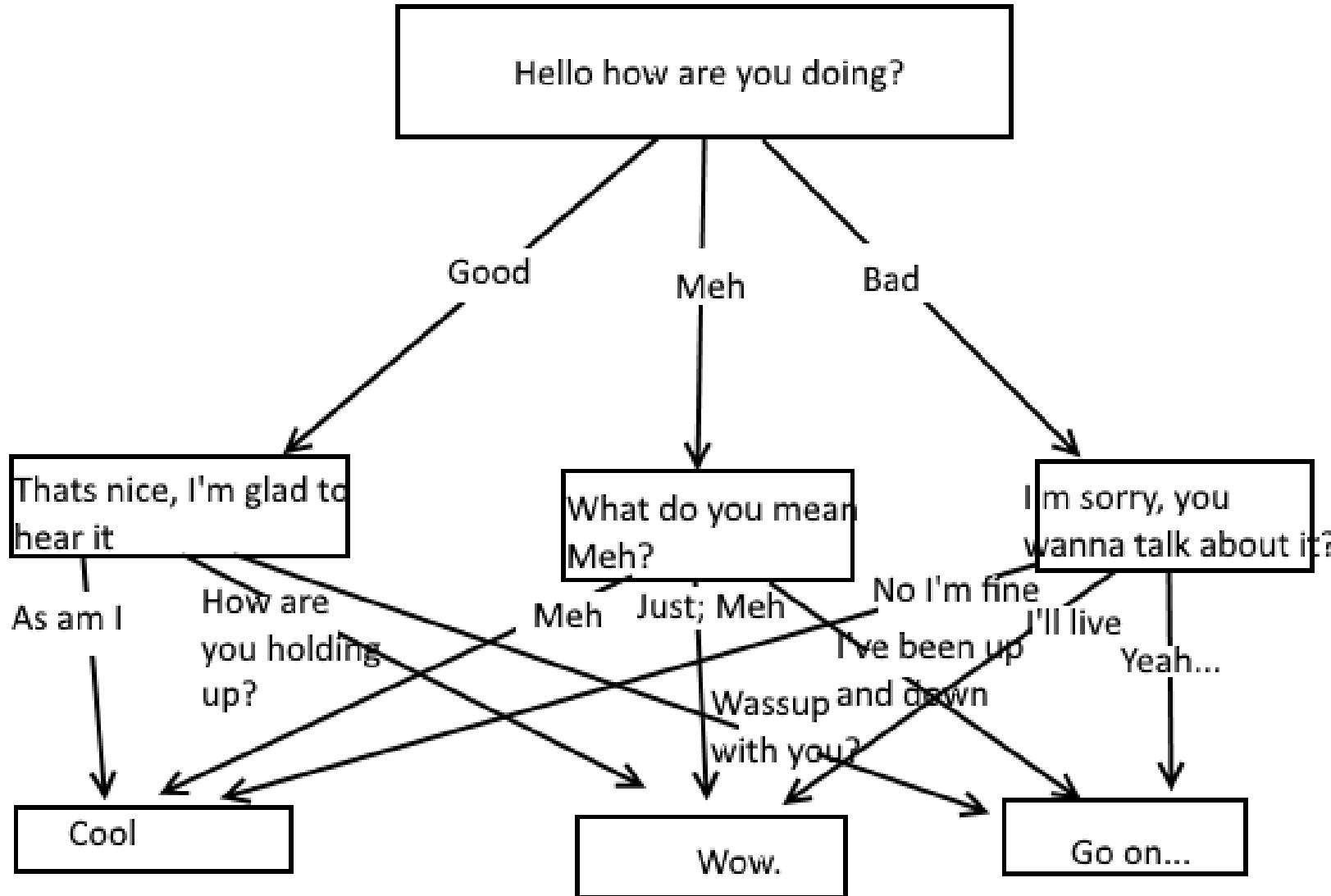


• Non-task-oriented Dialogue System

- **Chit-chat:** casual and non-goal-oriented.
- **Open domain and open ended**
- **Challenges:**
 - Coherence
 - Diversity
 - Engagement
 - ...
- **Ultimate goal:** to pass Turing Test



• Template-based (Rule-based) Solution

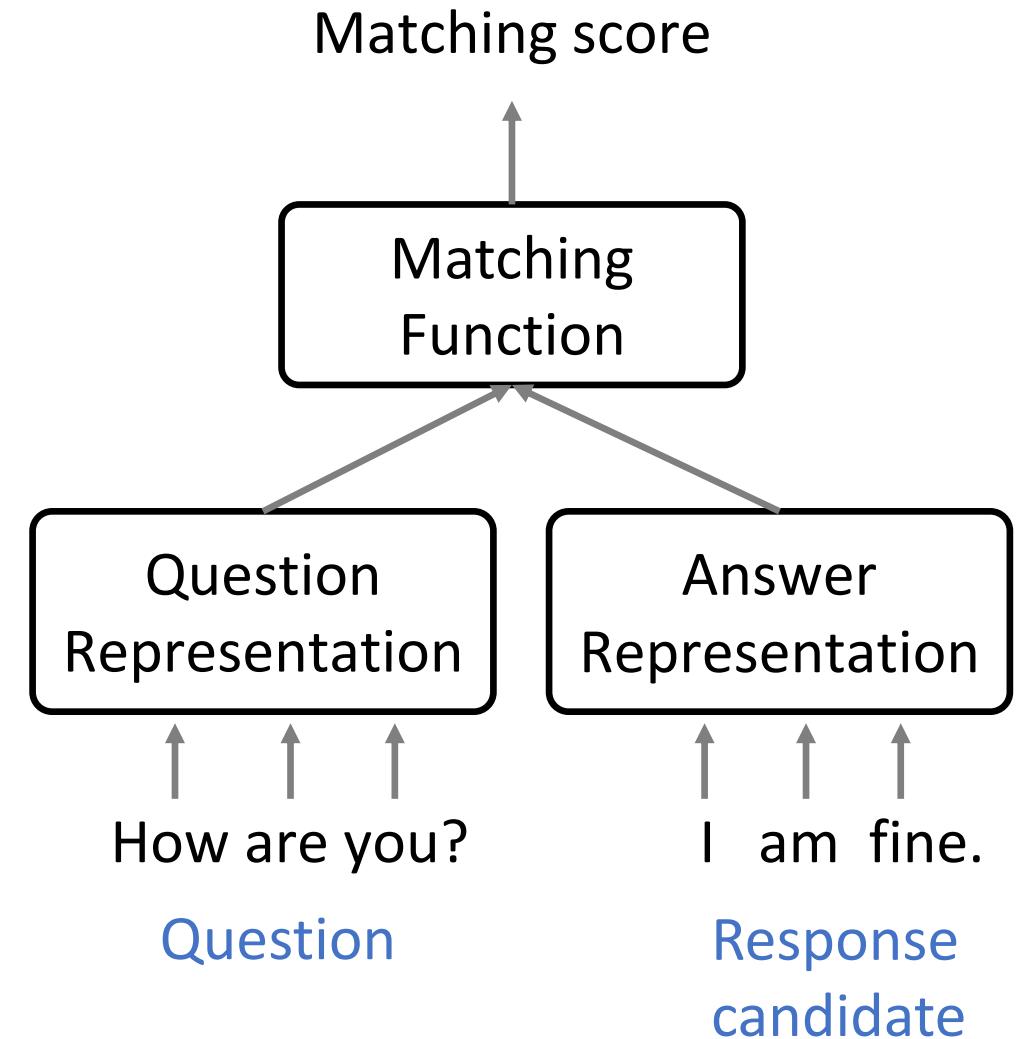


- **Unscalable**: require human labor
- **Inflexible**: hard to adopt to unseen topic

• Retrieval-based Solution

❑ Assumption:

- A large candidate response set such that all input utterances can get a proper response.



• Generation-based Solution -- Classical Sequence to Sequence

• Challenges:

• Blandness

- Basic models tend to generate generic responses like "I see" and "OK".

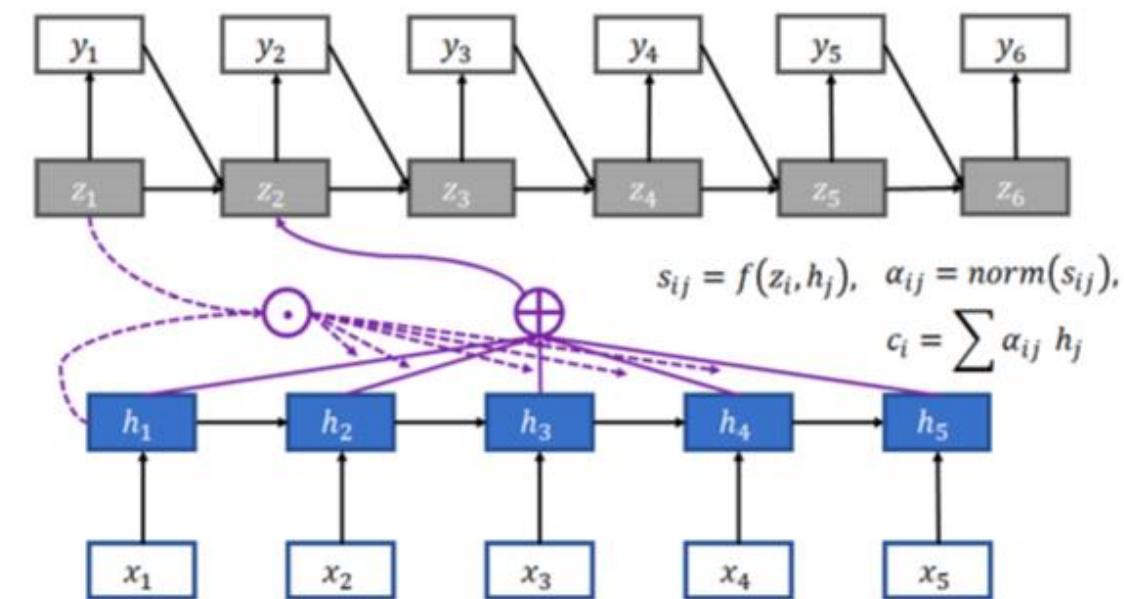
• Consistency

- Logical self-consistent across multiple turns, e.g., persona, sentiment

• Lack of Knowledge

- Typical sequence-to-sequence models only mimic surface level sequence ordering patterns without understanding world knowledges deeply.

A Basic Model:
Encoder-Attention-Decoder



• Blandness: VAE-based solution

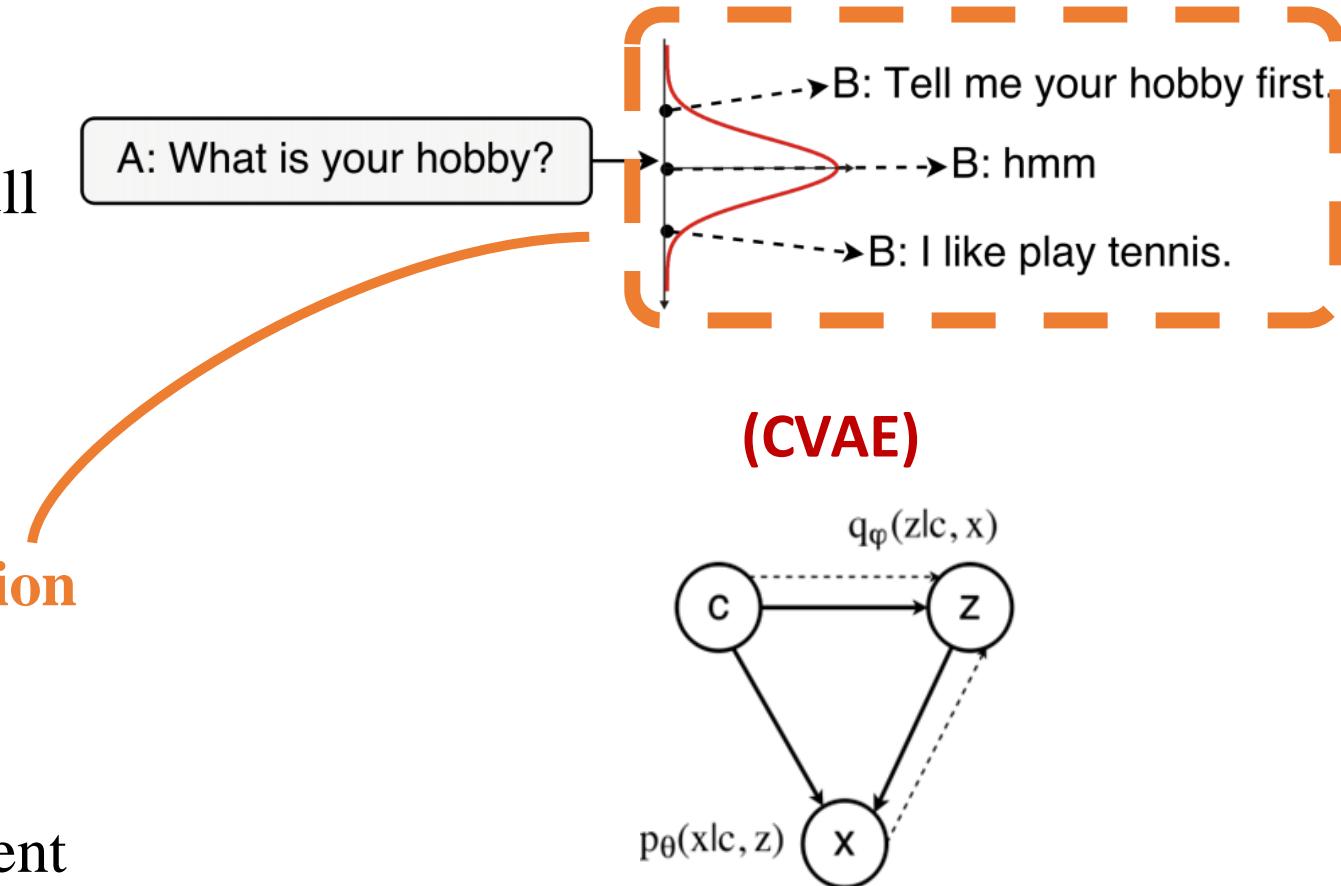
• Problem in chatbot:

- The lack of diversity: often generate dull and generic response.

• Solution:

- Using latent variables to learn a **distribution over potential conversation actions.**
- Using **Conditional Variational Autoencoders (CVAE)** to infer the latent variable.

Zhao et al. "Learning Discourse-level Diversity for Neural Dialog Models using Conditional Variational Autoencoders?"(ACL' 17)



- c : dialog history information
- x : the input user utterance
- z : latent vector of distribution of intents
- y : linguistic feature knowledge

• Consistency: Persona chat

• Motivation:

- The lack of a consistent personality
- A tendency to produce non-specific answers like “I don’t know”

• Solution: endowing machines with a configurable and consistent persona (profile), making chats condition on:

1. The machine’ own given profile information.
2. Information about the person the machine is talking to.

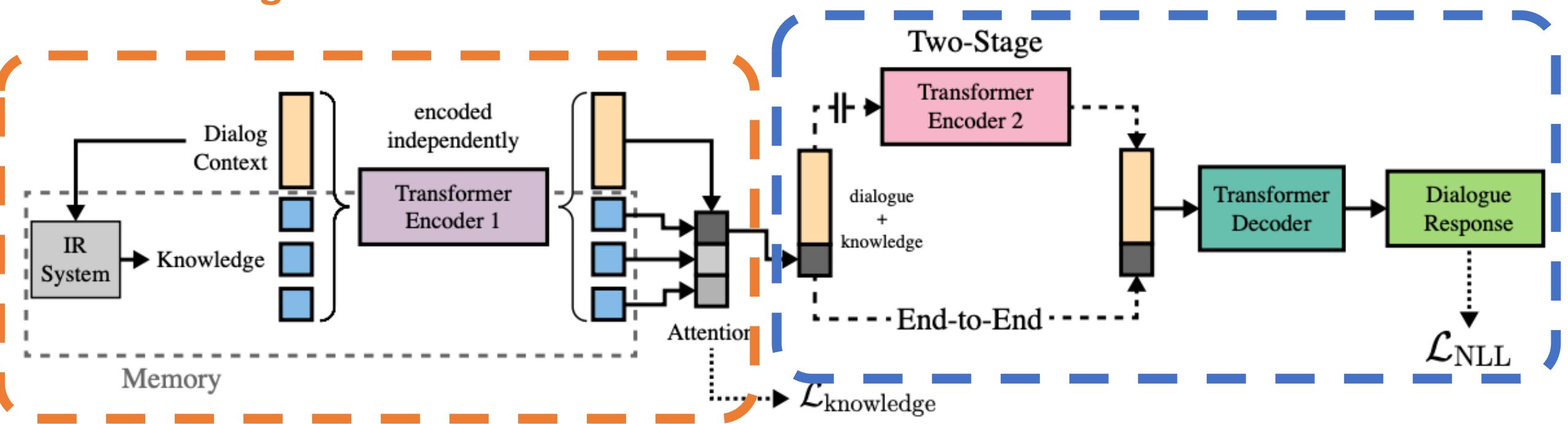
Persona of two interlocutors	
Persona 1	Persona 2
I like to ski	I am an artist
My wife does not like me anymore	I have four children
I have went to Mexico 4 times this year	I recently got a cat
I hate Mexican food	I enjoy walking for exercise
I like to eat cheetos	I love watching Game of Thrones
[PERSON 1:] Hi	
[PERSON 2:] Hello ! How are you today ?	
[PERSON 1:] I am good thank you , how are you.	
[PERSON 2:] Great, thanks ! My children and I were just about to watch Game of Thron	
[PERSON 1:] Nice ! How old are your children?	
[PERSON 2:] I have four that range in age from 10 to 21. You?	
[PERSON 1:] I do not have children at the moment.	
[PERSON 2:] That just means you get to keep all the popcorn for yourself.	
[PERSON 1:] And Cheetos at the moment!	
[PERSON 2:] Good choice. Do you watch Game of Thrones?	
[PERSON 1:] No, I do not have much time for TV.	
[PERSON 2:] I usually spend my time painting: but, I love the show.	

- Lack of background knowledge: Knowledge grounded dialogue response generation -- Text

- Solution: Knowledge retrieval from texts (e.g., Wikipedia) into dialogue responses

Knowledge retrieval module

Response generated by integrating knowledge

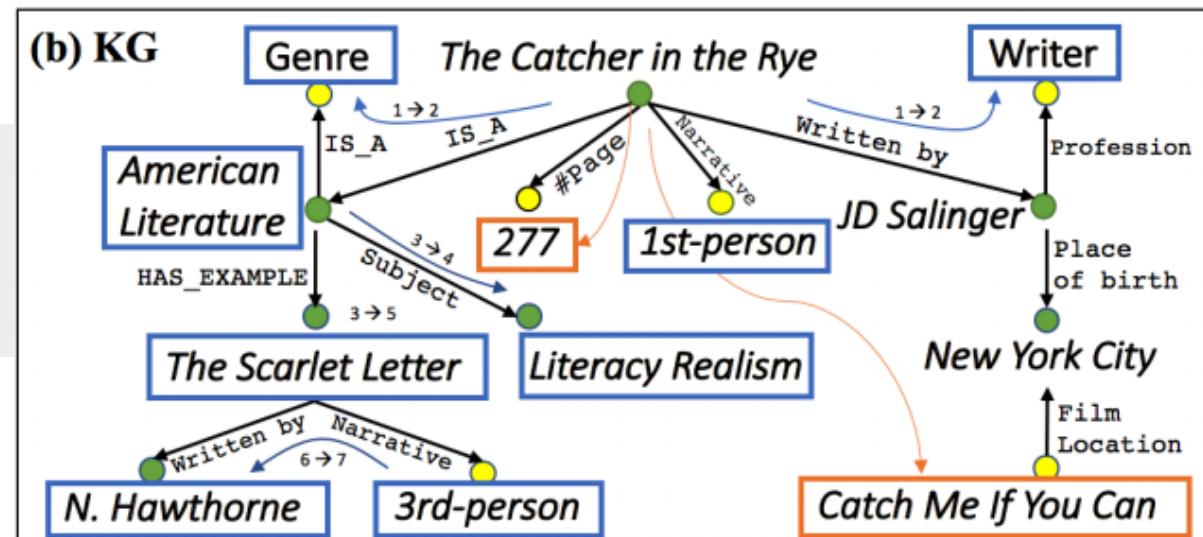
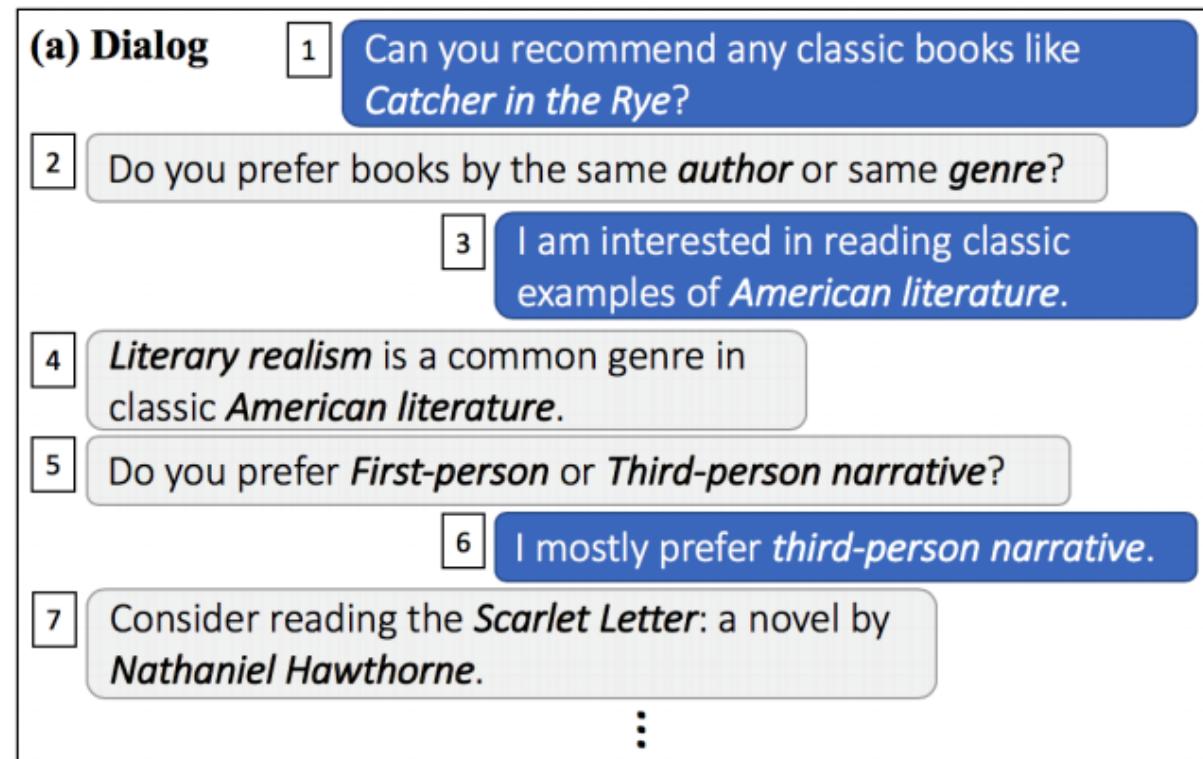


- Lack of background knowledge: Knowledge grounded dialogue response generation -- KG

- Solution: **Walking** within a large knowledge graph to
 - track dialogue states.
 - to guide dialogue planning

Blue arrow: walkable paths led to engaging dialogues
Orange arrow: non-ideal paths that never mentioned
 (Should be pruned)

Moon et al. "OpenDialKG: Explainable Conversational Reasoning with Attention-based Walks over Knowledge Graphs" (ACL' 19)



• Tutorial Outline

- ❑ A Glimpse of Dialogue System
- ❑ Four research directions in conversational recommendation system
 - ❑ **Question Driven Approaches**
 - ❑ Multi-turn Conversational Recommendation Strategy
 - ❑ Dialogue Understanding and Generation
 - ❑ Exploitation-Exploration Trade-offs for Cold Users
 - ❑ Summary of Formalizations and Evaluations

• System Ask – User Respond (SAUR) - Formalization

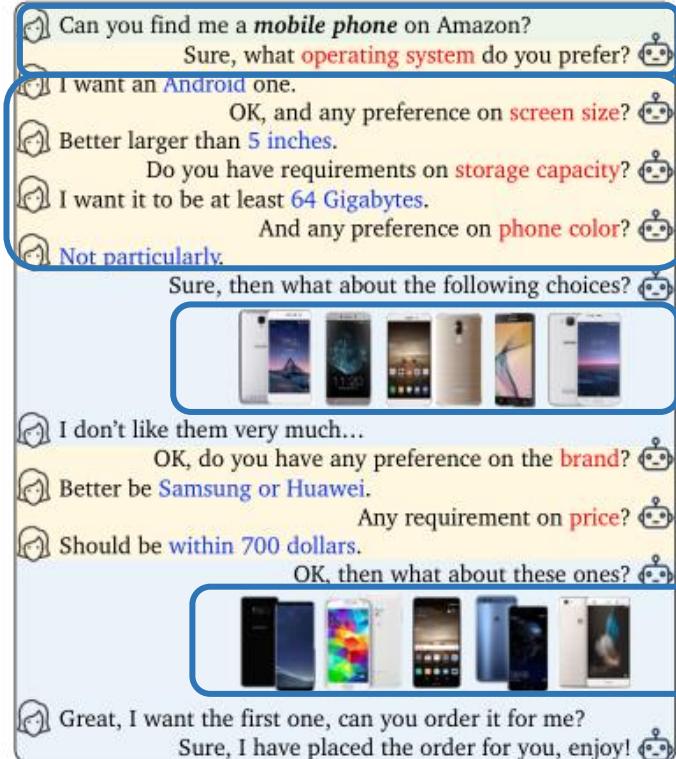


Figure 1: Example for conversational search in e-commerce product search or recommendation scenario (best in color).

Initial request

Get feedback

Feels confident

1. Initiation

User initiates a conversation

2. Conversation

Asks the user preferences on product aspects

3. Display

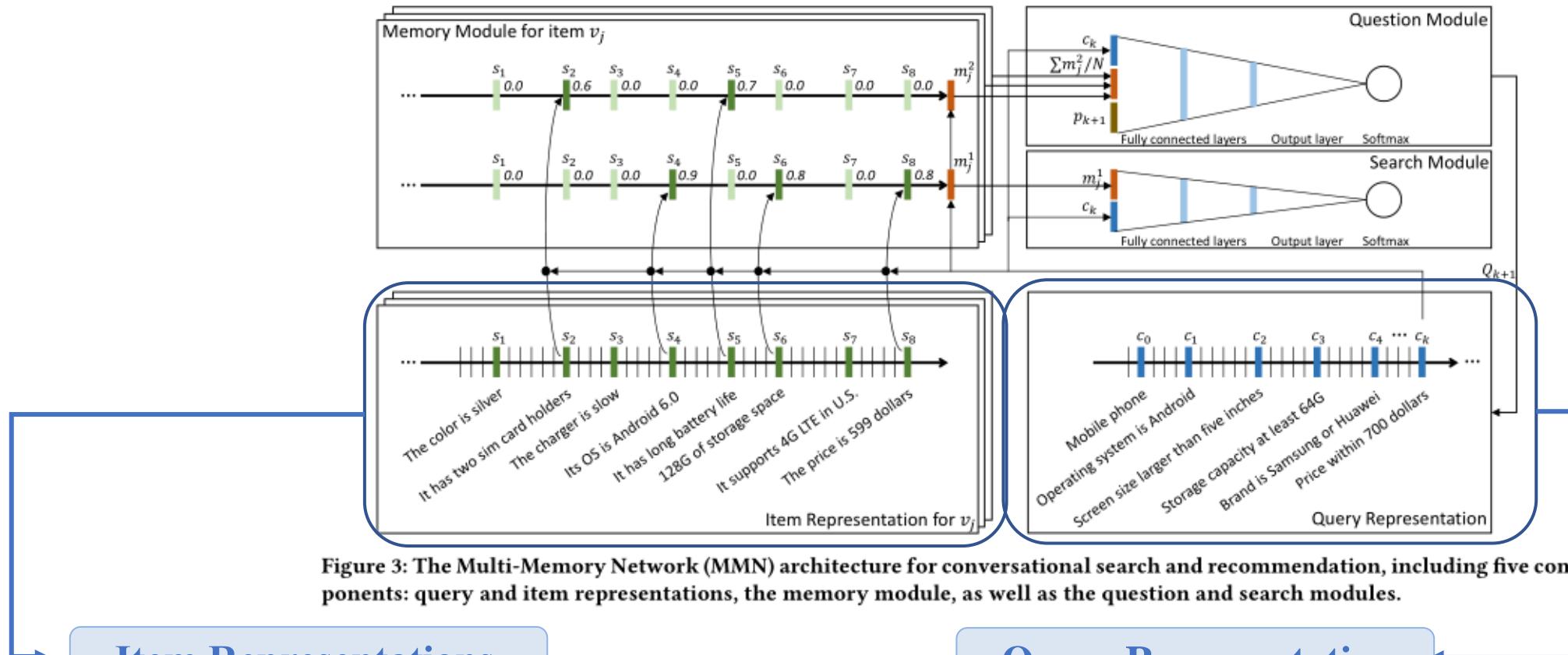
Display product to the user

Research Question -- Given the requests specified in dialogues, the system needs to predict:

1. What questions to ask
2. What items to recommend

three stages

• SAUR – Method -- Representation



Item Representations

- Apply a **gated recurrent unit** (GRU) on the text description of each item
- The **hidden states of each sentence** as the representation of $T_j(s_1, s_2 \dots s_\tau)$

Query Representation

- Also a **gated recurrent unit** (GRU)
- Query sequence $c_1, c_2 \dots$ is extracted in conversations

• SAUR - Method

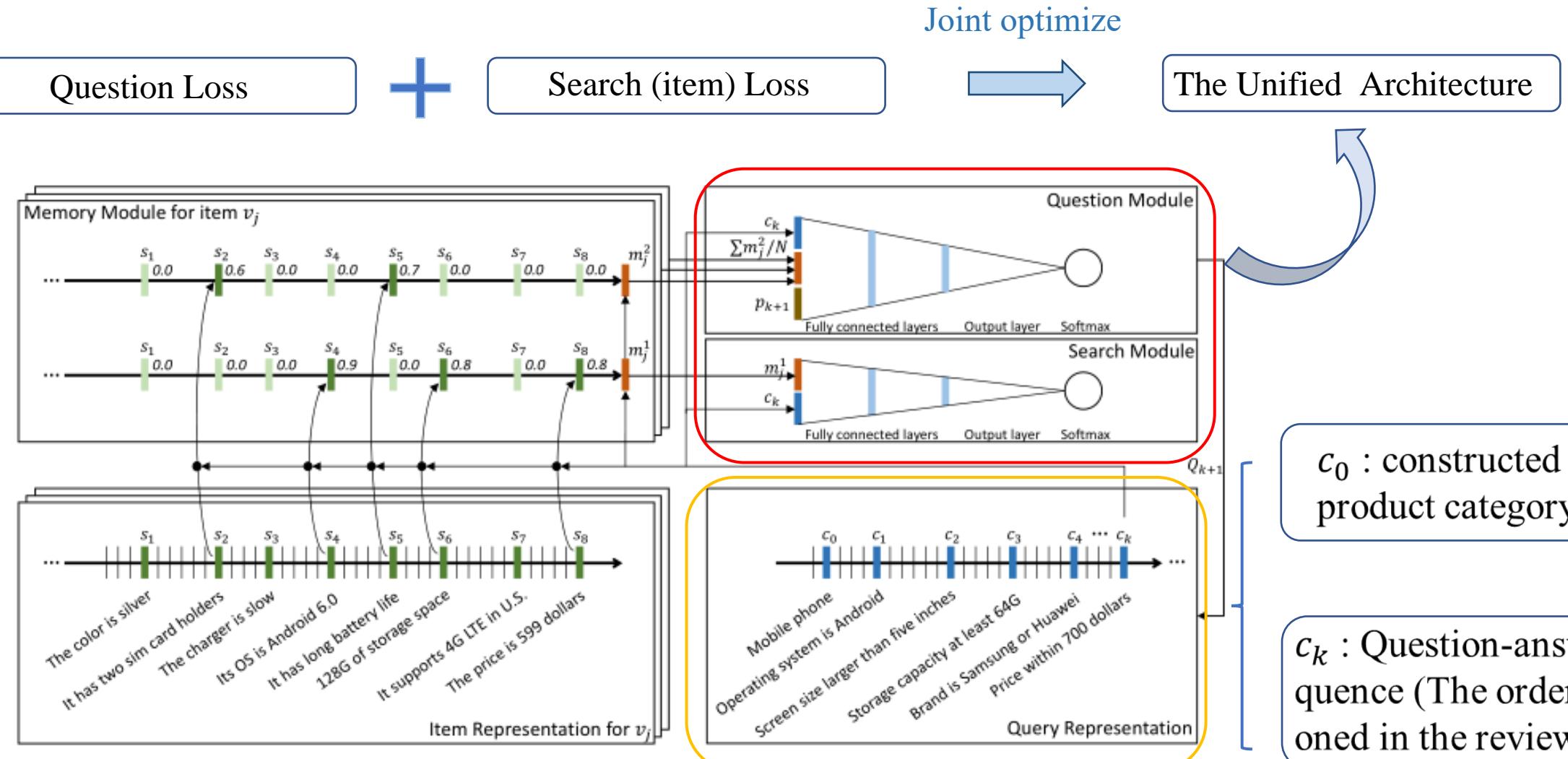
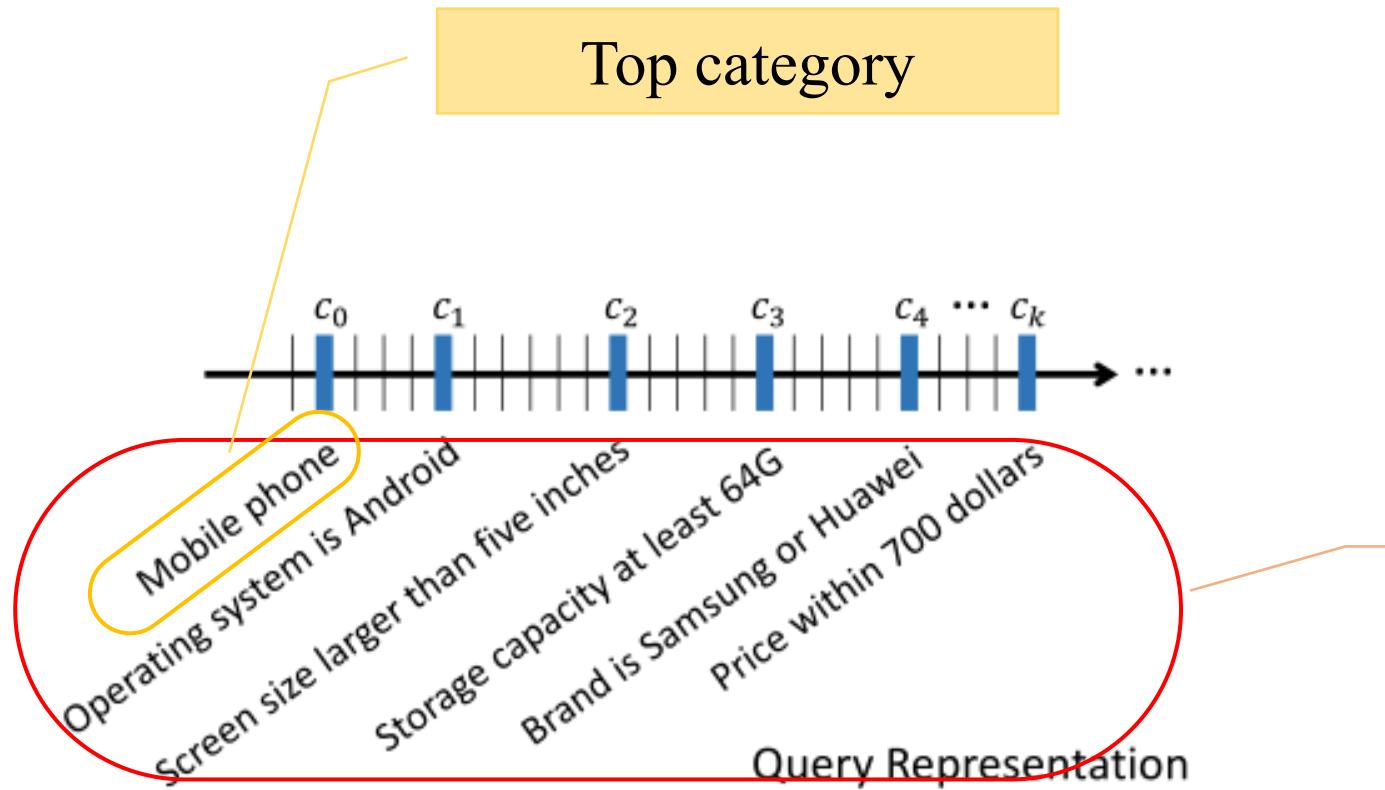


Figure 3: The Multi-Memory Network (MMN) architecture for conversational search and recommendation, including five components: query and item representations, the memory module, as well as the question and search modules.

• SAUR - Evaluation

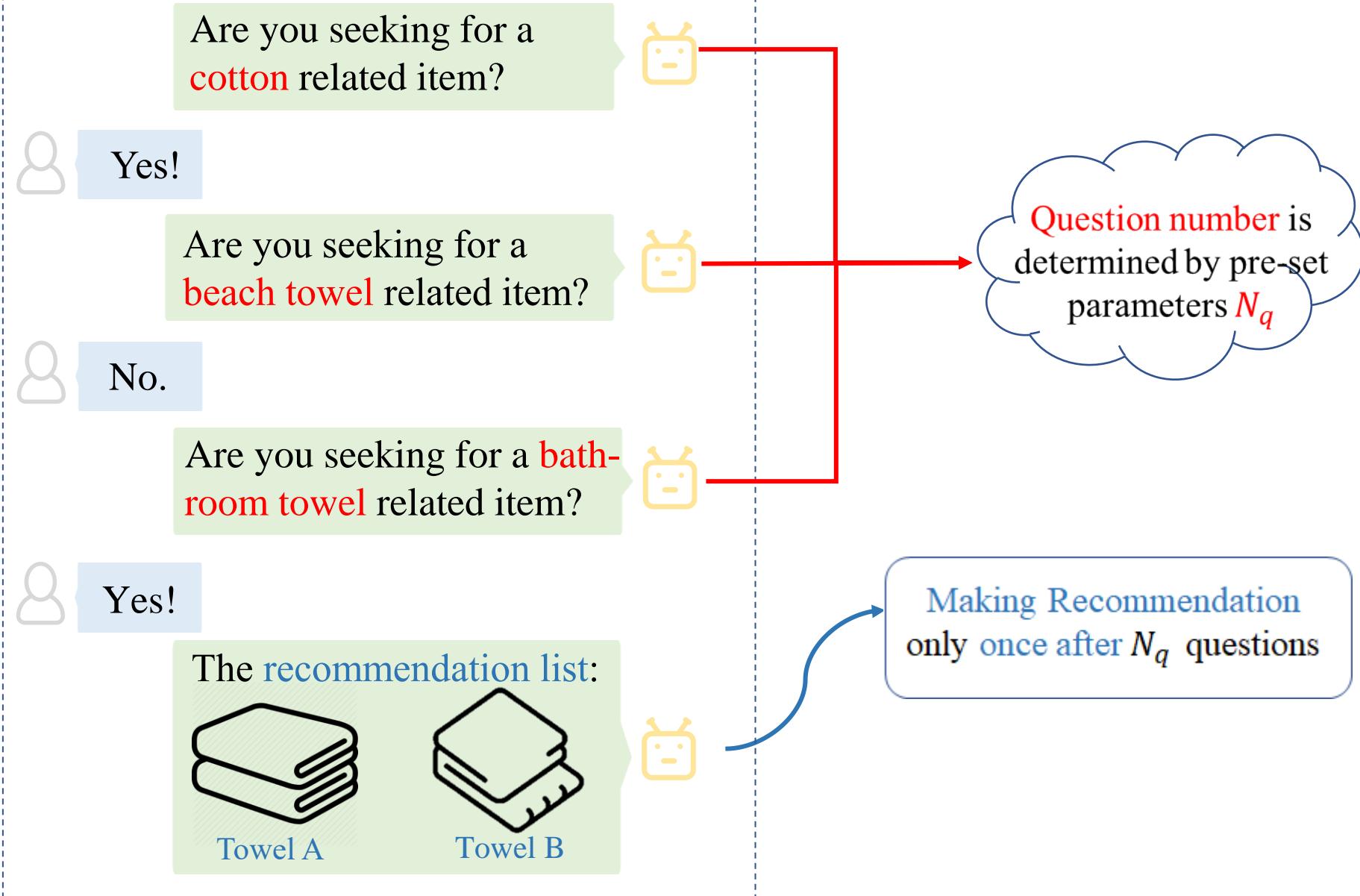
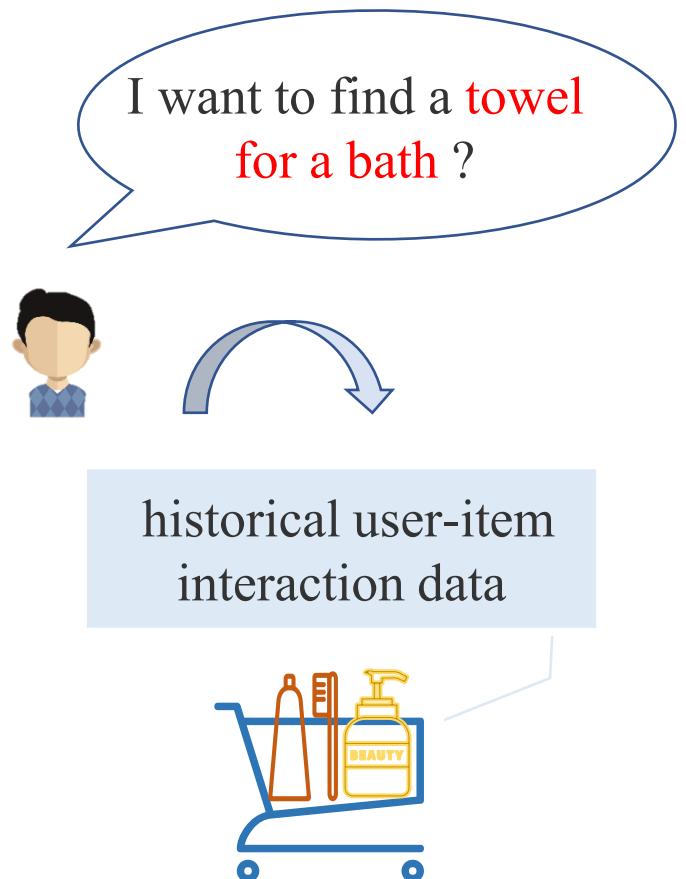


Evaluation Criteria:

1. Query prediction
2. Item prediction (e.g., NDCG)

User's review

• Question-based recommendation(Qrec) - Formalization



• Qrec - Method -- Offline and Online Optimization

Latent Factor Recommendation

Y_{ij} : the number of attributes of item j that satisfy the user i in current conversation.

For example, if a user specify [cotton] and [towel], but the item only gets [cotton], $Y=1$.

$$\begin{aligned} & -\log p(U, V | R, Y, \Theta) \\ & \propto \frac{1}{2} \sum_{i,j \in R} (R_{ij} - p^T(u_i \circ v_j))^2 + \frac{\gamma}{2} \sum_{i,j \in Y} (Y_{ij} - q^T(u_i \circ v_j))^2 + \\ & \quad \sum_{i=1}^M \frac{\lambda_u}{2} \|u_i\|_2^2 + \sum_{j=1}^N \frac{\lambda_v}{2} \|v_j\|_2^2 + \frac{\lambda_p}{2} \|p\|_2^2 + \frac{\lambda_q}{2} \|q\|_2^2, \end{aligned} \quad (2)$$

Offline Optimization

Online Optimization
(feedback from user,
(i.e. Y))

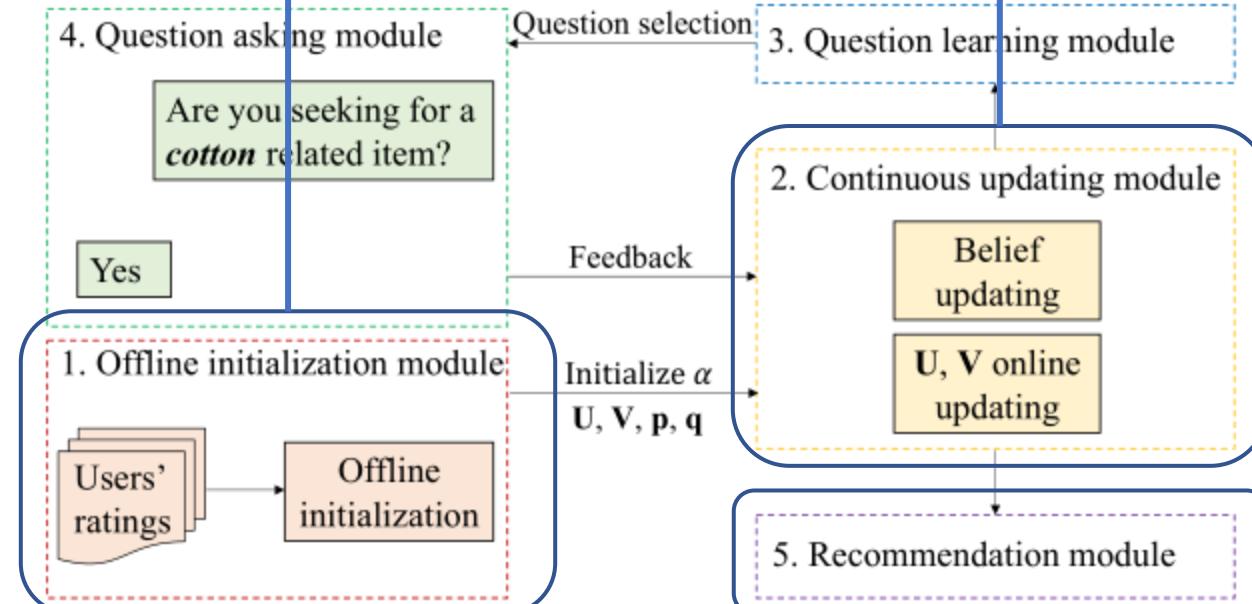
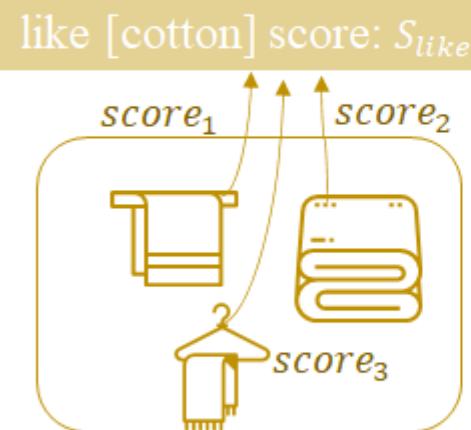


Figure 1: Framework of our proposed question-based recommendation model, Qrec. **Cotton** is an extracted entity (informative term), U, V, p, q are model variables, and α is a hyper-parameter of user belief.

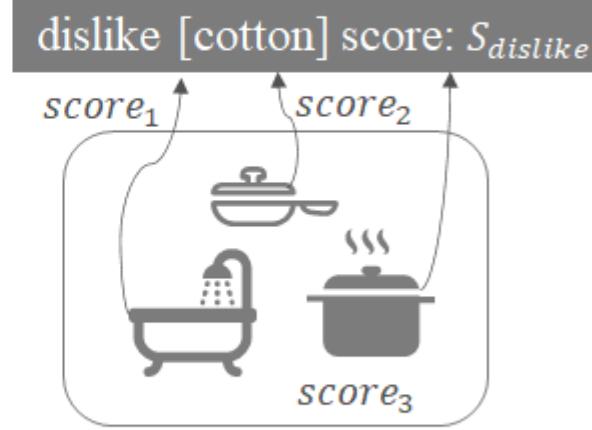
• Qrec - Method -- Choosing Questions to Ask

Attribute Choosing criteria: Finding the most uncertain [attribute] to ask.

[cotton] preference confidence : $|S_{like} - S_{dislike}|$



[cotton] related
item set



[cotton] unrelated
item set

The smaller the preference confidence indicate the more uncertain attribute.

$$score_i \propto (UV^T, Y_i)$$

• Qrec - Evaluation

Evaluation Measures:
recall@5, MRR, NDCG
only on items!

No questions are evaluated,
but if question asking strategy
is bad, the item
recommendation results will
not be good.

Dataset: *Amazon product dataset*

- Using TAGME (an entity linking tool) to find the entities in the product description page as the attributes.

Item Name: “Cotton Hotel spa Bathroom Towel”

Item Attributes: [cotton, bathroom, hand towels]

Template-based
question

Are you seeking for a
cotton related item?



Yes!

Are you seeking for a
beach towel related item?



No.

Are you seeking for a **bath-**
room towel related item?



Yes!

The recommendation list:



Towel A



Towel B

simulate

• Question & Recommendation(Q&R) - Formalization

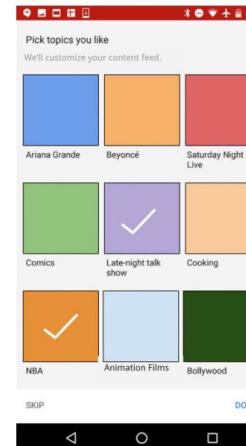
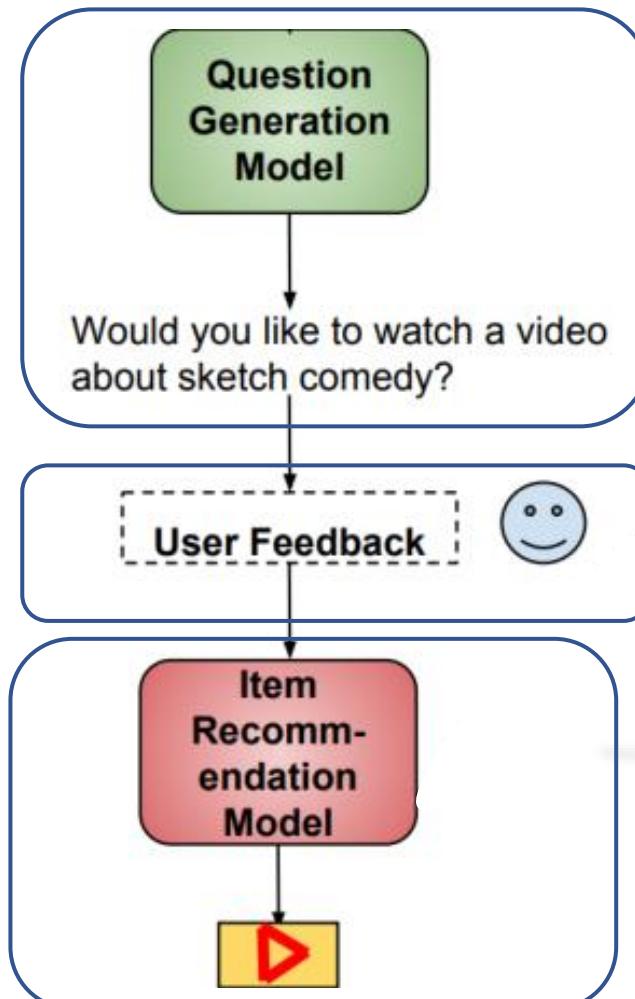


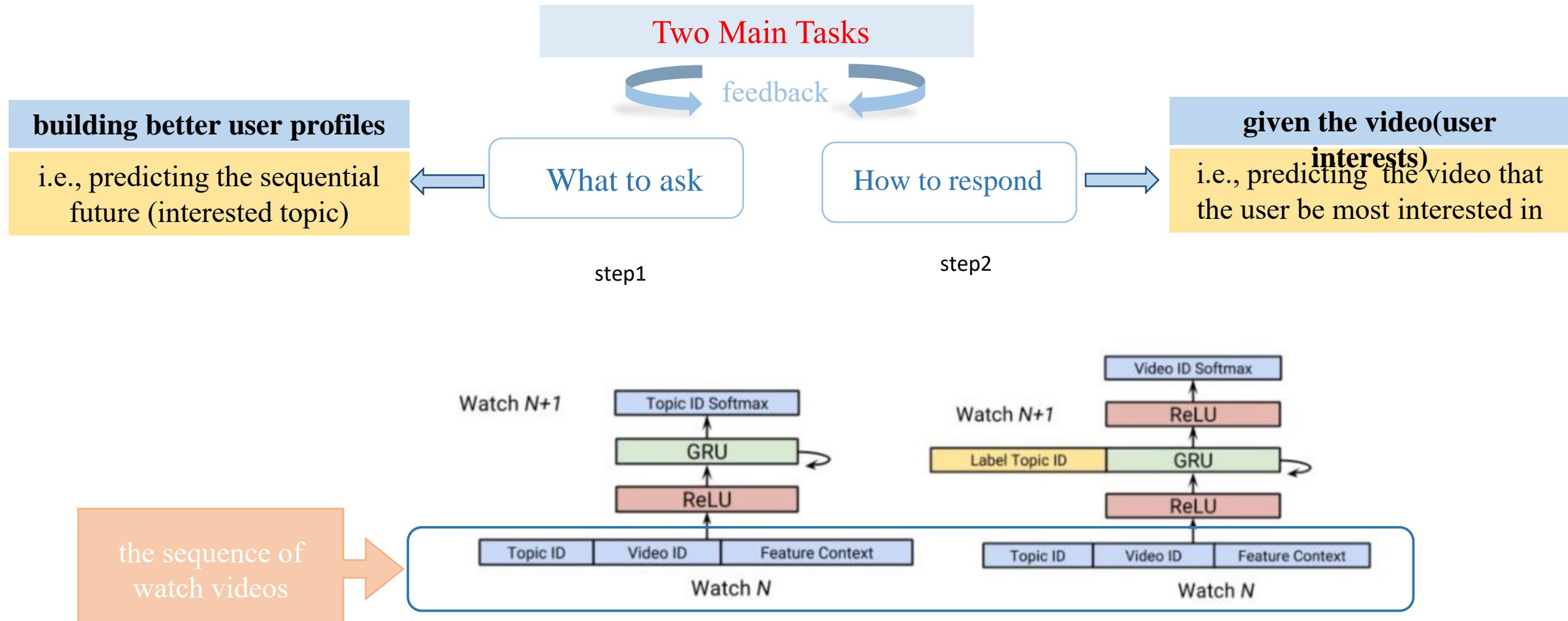
Figure 2: User Onboarding UI.

Positive-only type of feedback
(click topics)

User is prompted
to choose as many
topics as they like

Incorporates the user
feedback to improve
video recommendations

• Q&R - Method



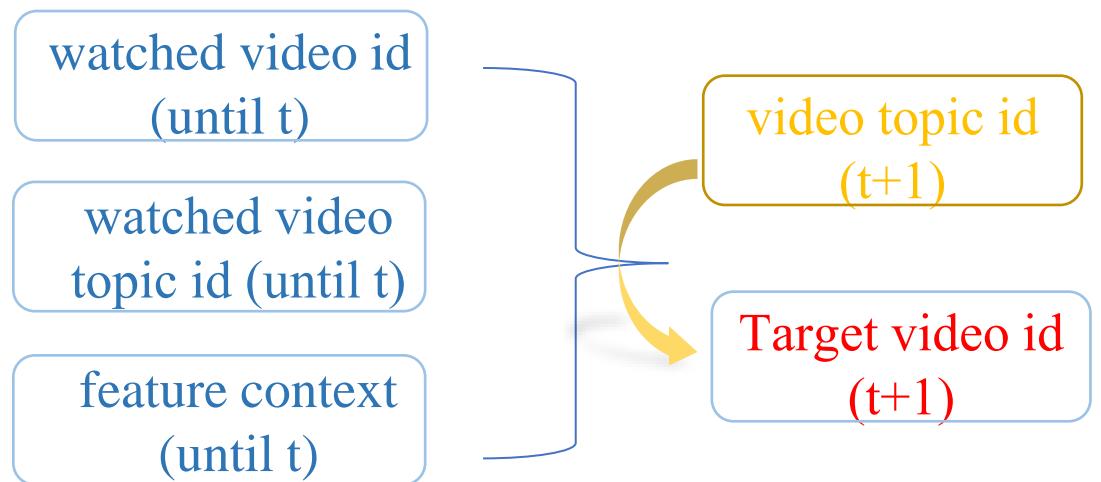
• Q&R - Evaluation

Offline Evaluation

Data

YouTube user watch sequences

1. The **watch sequence** of a user up until the previous to last step
2. The **video ID** and **topic ID** of the user's last watch event



Online Evaluation

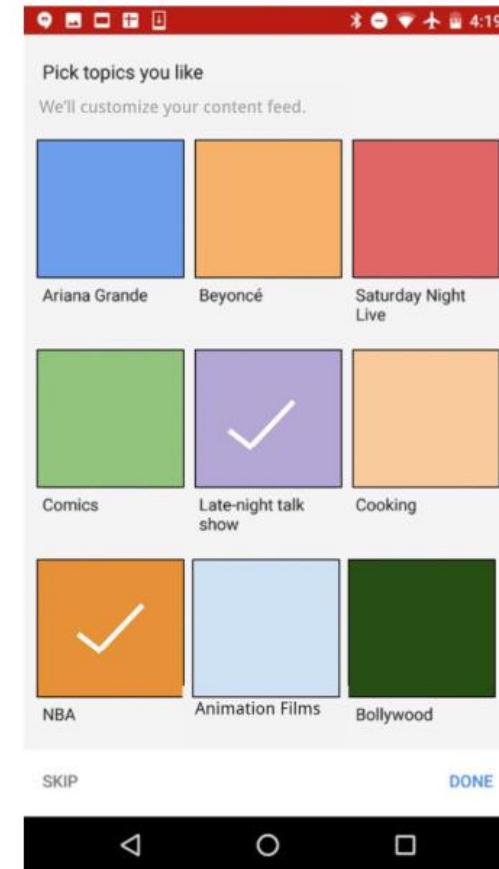


Figure 2: User Onboarding UI.

• Tutorial Outline

- ❑ A Glimpse of Dialogue System
- ❑ Four research directions in conversational recommendation system
 - ❑ Question Driven Approaches
 - ❑ **Multi-turn Conversational Recommendation Strategy**
 - ❑ Dialogue Understanding and Generation
 - ❑ Exploitation-Exploration Trade-offs for Cold Users
- ❑ Summary of Formalizations and Evaluations

• CRM - Formalization

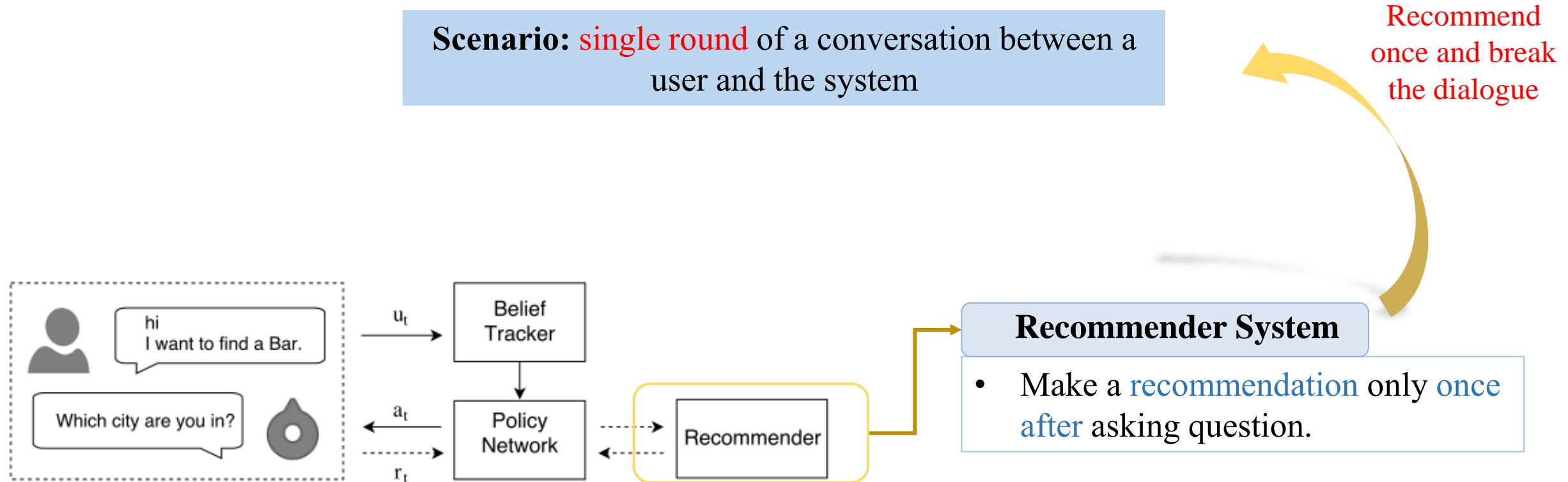


Figure 1: The conversational recommender system overview

• CRM - Method -- Dialogue Component

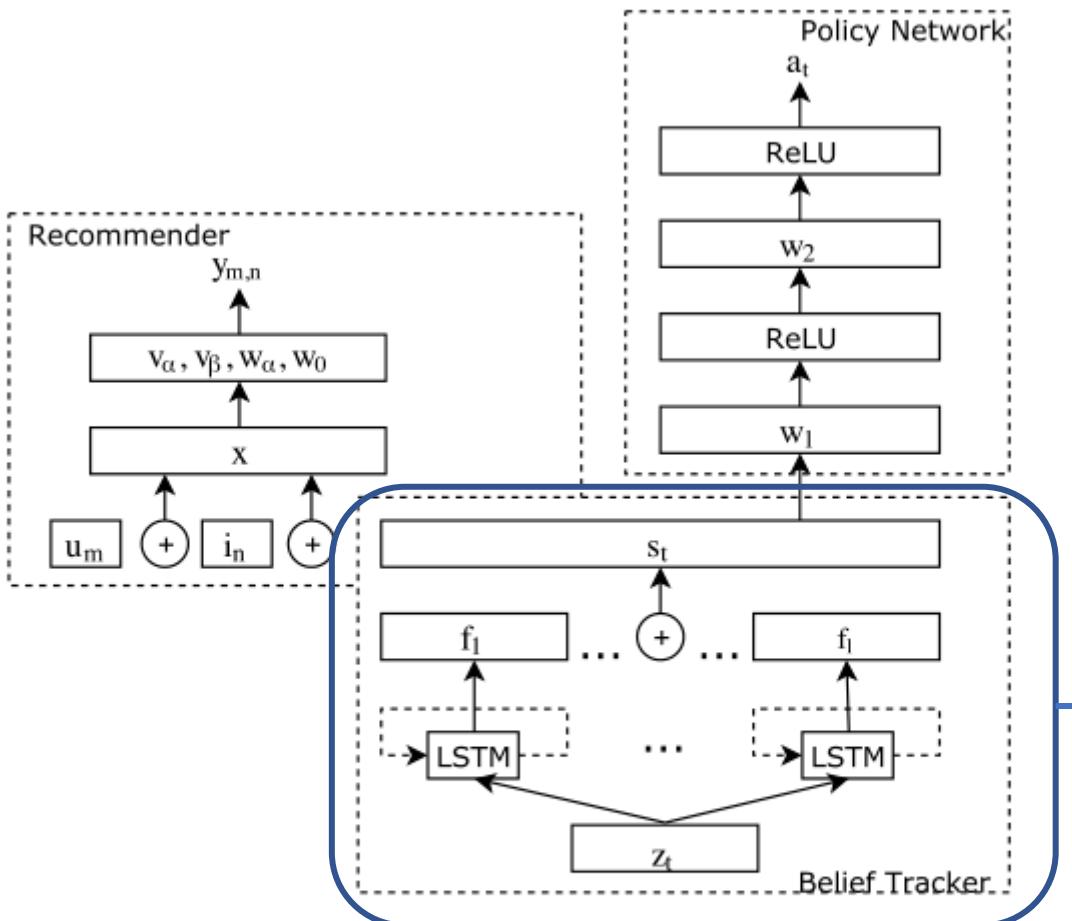
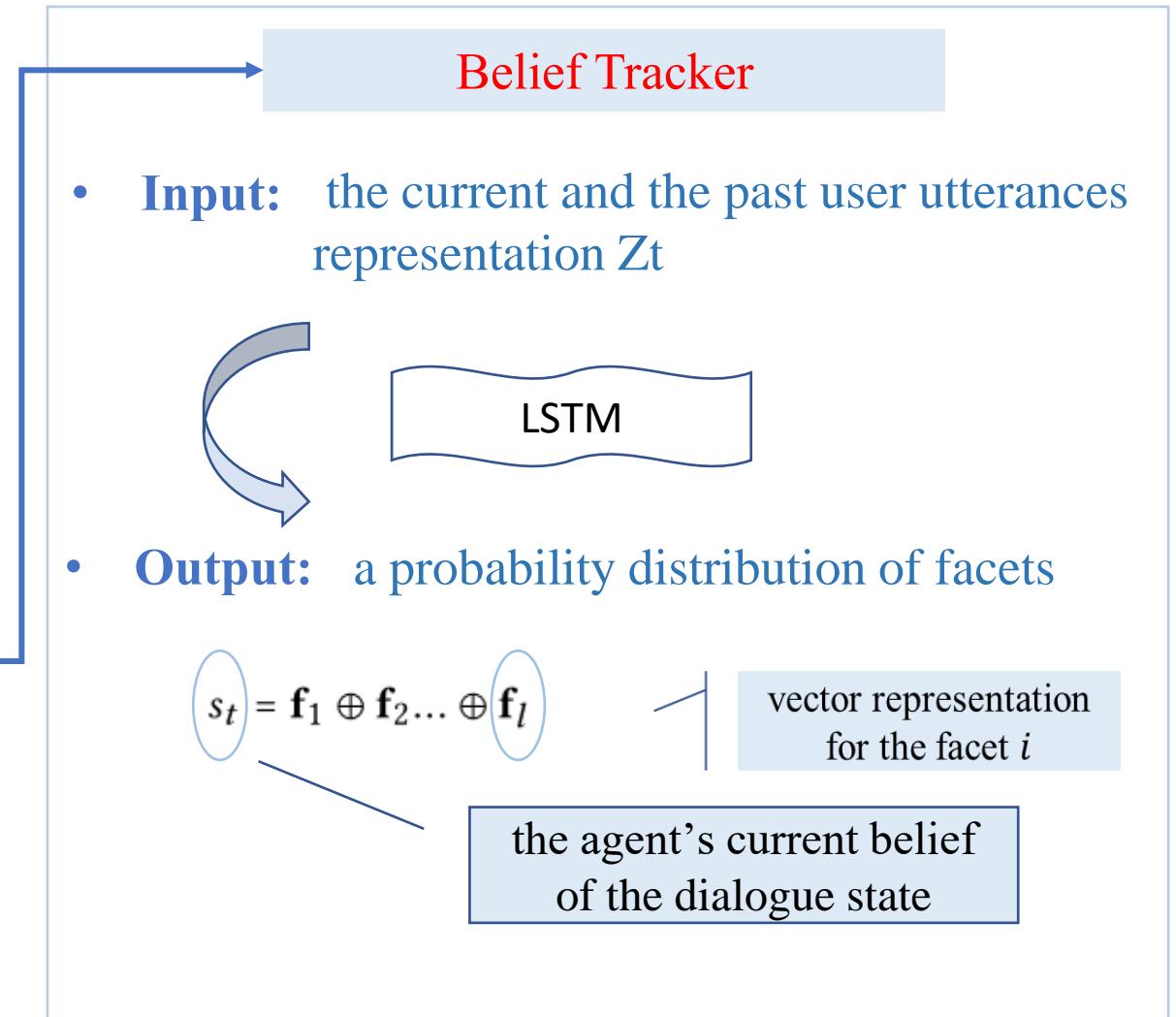


Figure 2: The structure of the proposed conversational recommender model. The bottom part is the belief tracker, the top left part is the recommendation model, and the top right part is the deep policy network.



• CRM - Method

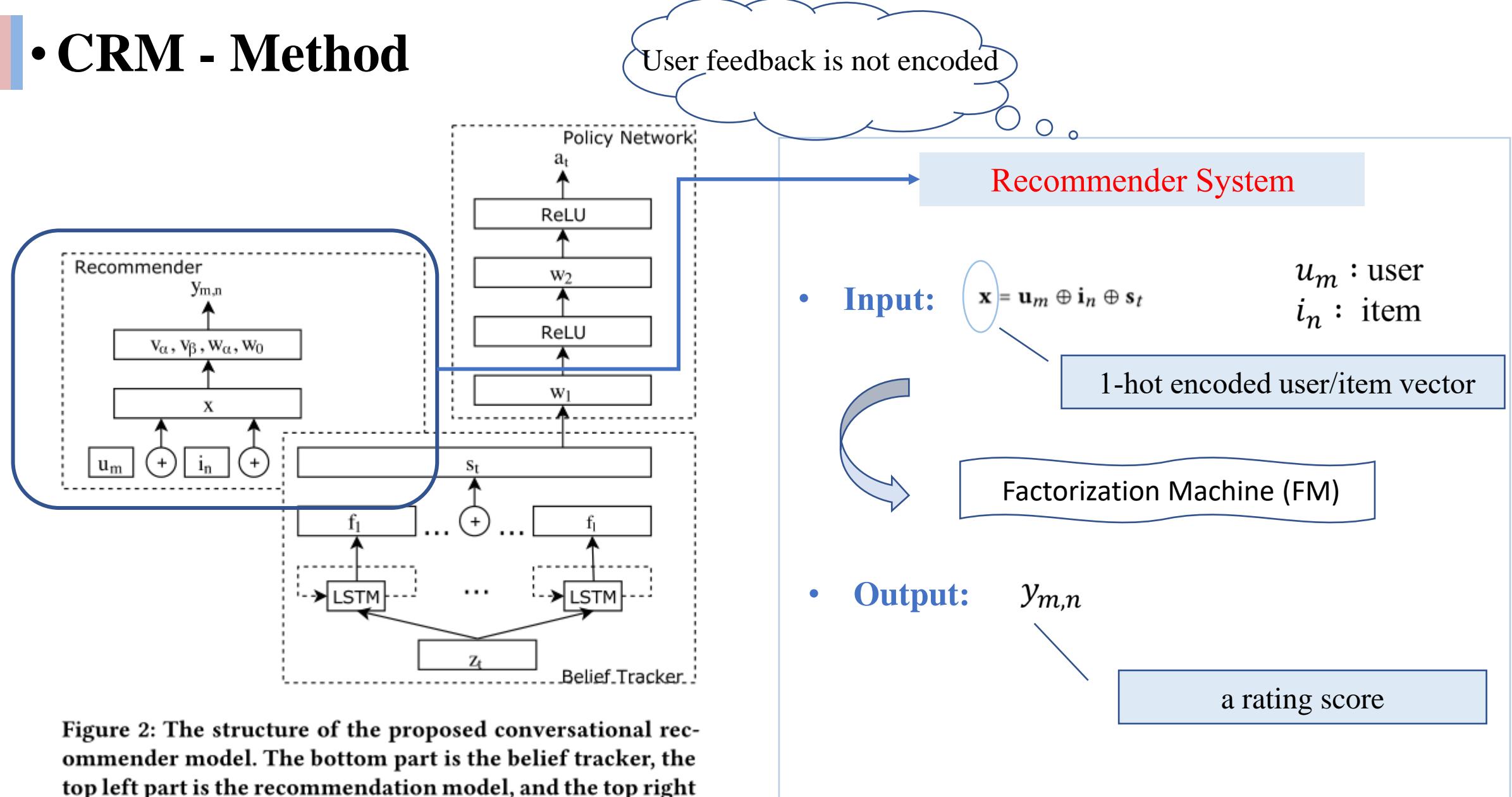
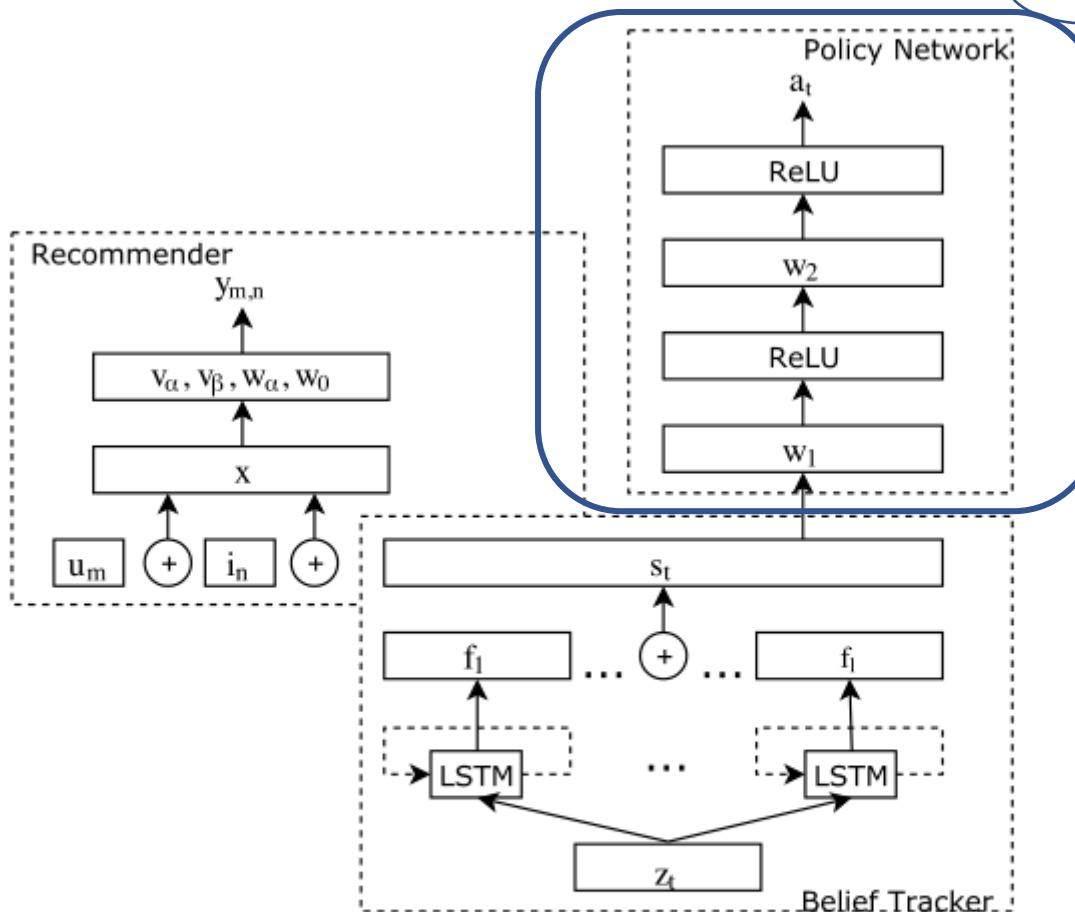


Figure 2: The structure of the proposed conversational recommender model. The bottom part is the belief tracker, the top left part is the recommendation model, and the top right part is the deep policy network.

• CRM - Method



Decisions based only on the belief tracker

Deep Policy Network

- **State:** $s_t = \{f_1 \oplus f_2 \dots \oplus f_l\}$.
- **Action**: $\{a_1, a_2, \dots, a_l\}$, request the value of a facet
make a personalized recommendation
- **Reward**: benefit/penalty the agent gets from interacting with its environment
- **Policy:** $\pi(a_t | s_t)$, two fully connected layers as the policy network

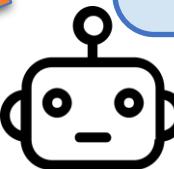
Adopt the policy gradient method of reinforcement learning

• CRM - Evaluation

User Simulation



Yelp (the restaurants and food data)



(city="Italian", category="San Diego")

I'm looking for Italian food in San Diego.

Which state are you in?

I'm in California. (state="CA")

Which price range do you like?

Low price (price_range="cheap")

What rating range do you want?

3.5 or higher. (rating_range>="3.5")

Do you want “Small Italy Restaurant”?

thank you!

Item Name: “Small Italy Restaurant”

Item Attributes: [Italian, San Diego, California, cheap, rating>=3.5]

Evaluation Metrics

Evaluation Matrices:

- SR @ k (Success rate at k-th turn)
- AT (Average Turns)

$$SR = \frac{\#successful\ dialogues}{\#dialogues} \cdot 100\%$$

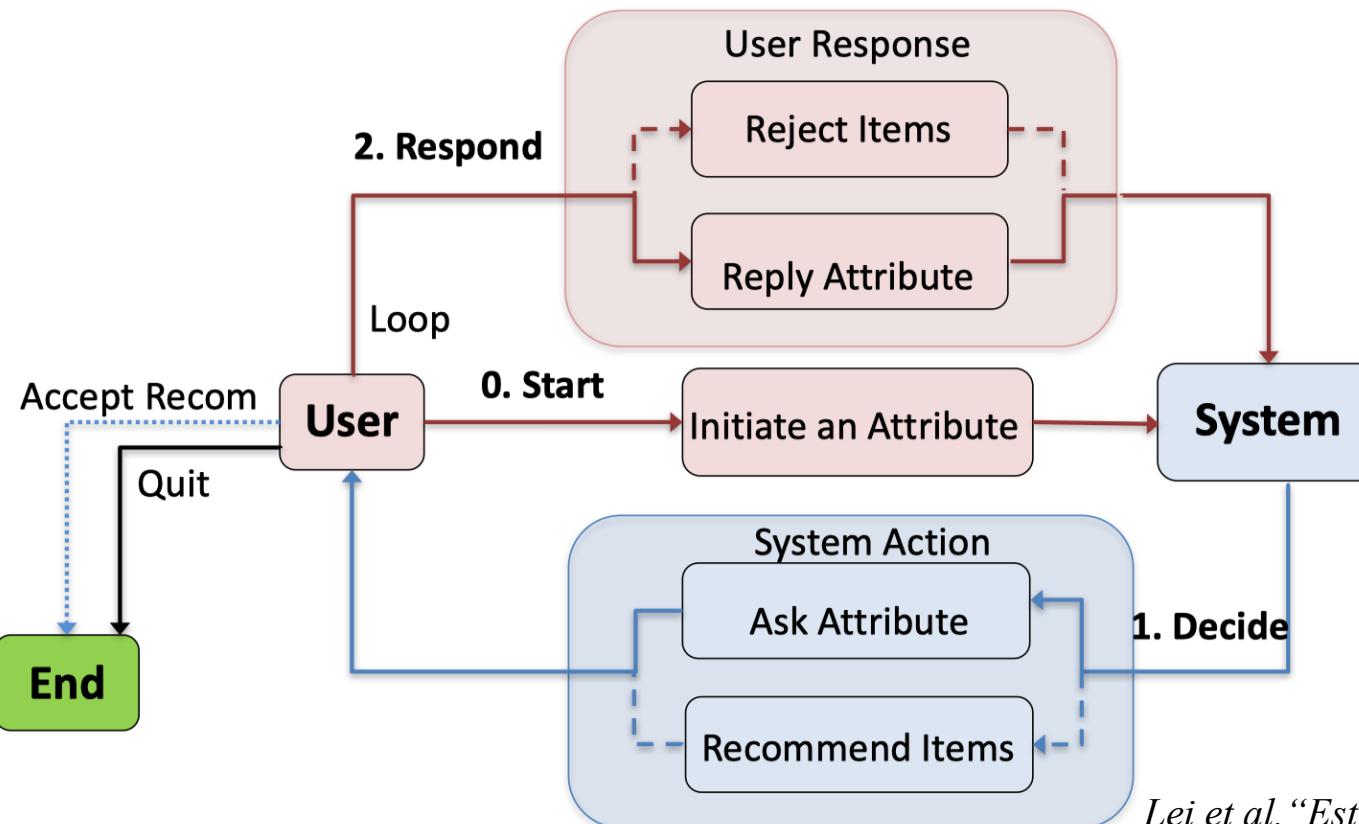
$$AT = \overline{dialogue\ length.}$$

• Estimation–Action–Reflection(EAR) - Formalization

Workflow of Multi-round Conversational Recommendation (MCR)

Objective:

Recommend desired items to user in shortest turns



- Key Research Questions
 - 1. What item/attribute to recommend/ask?
 - 1. Strategy to ask and recommend?
 - 1. How to adapt to user's online feedback?

Lei et al. "Estimation–Action–Reflection: Towards Deep Interaction Between Conversational and Recommender Systems" (WSDM'20)

• EAR - Method -- What Item to Recommend and What Attribute to Ask

Method: Attribute-aware FM for Item Prediction and Attribute Preference Prediction

Notation	Meaning
p	A given attribute
u	User embedding
\mathcal{P}_u	User's known preferred attributes

Notation	Meaning
(Neg. 1) $\mathcal{V}_u^- := \mathcal{V} \setminus \mathcal{V}_u^+$	The ordinary negative sample as in standard BPR.
(Neg. 2) $\widehat{\mathcal{V}_u} := \mathcal{V}_{cand} \setminus \mathcal{V}_u^+$	\mathcal{V}_{cand} is the set of candidate items satisfying user's preferred attributes.
$\mathcal{D}_1 := \{(u, v, v') v' \in \mathcal{V}_u^-\}$	Paired sample for first kind of negative sample
$\mathcal{D}_2 := \{(u, v, v') v' \in \widehat{\mathcal{V}_u}\}$	Paired sample for second kind of negative sample

$$\hat{y}(u, v, \mathcal{P}_u) = \mathbf{u}^T \mathbf{v} + \sum_{p_i \in \mathcal{P}_u} \mathbf{v}^T \mathbf{p}_i$$

Score function for item prediction

$$L_{item} = \sum_{(u, v, v') \in \mathcal{D}_1} -\ln \sigma(\hat{y}(u, v, \mathcal{P}_u) - \hat{y}(u, v', \mathcal{P}_u)) + \sum_{(u, v, v') \in \mathcal{D}_2} -\ln \sigma(\hat{y}(u, v, \mathcal{P}_u) - \hat{y}(u, v', \mathcal{P}_u)) + \lambda_{\Theta} \|\Theta\|^2$$

ordinary negative example
The items satisfying the specified attribute but still are not clicked by the user

• EAR - Method -- What Item to Recommend and What Attribute to Ask

Method: Attribute-aware FM for Item Prediction and Attribute Preference Prediction

Notation	Meaning
p	A given attribute
u	User embedding
\mathcal{P}_u	User's known preferred attributes

Notation	Meaning
(Neg. 1) $\mathcal{V}_u^- := \mathcal{V} \setminus \mathcal{V}_u^+$	The ordinary negative sample as in standard BPR.
(Neg. 2) $\widehat{\mathcal{V}_u} := \mathcal{V}_{cand} \setminus \mathcal{V}_u^+$	\mathcal{V}_{cand} is the set of candidate items satisfying user's preferred attributes.
$\mathcal{D}_1 := \{(u, v, v') v' \in \mathcal{V}_u^-\}$	Paired sample for first kind of negative sample
$\mathcal{D}_2 := \{(u, v, v') v' \in \widehat{\mathcal{V}_u}\}$	Paired sample for second kind of negative sample

$$\hat{g}(p|u, \mathcal{P}_u) = u^T p + \sum_{p_i \in \mathcal{P}_u} \mathbf{P}^T \mathbf{P}_i$$

Score function for attribute preference prediction

$$L_{attr} = \sum_{(u, p, p') \in \mathcal{D}_3} -\ln \sigma(\hat{g}(p|u, \mathcal{P}_u) - \hat{g}(p'|u, \mathcal{P}_u)) + \lambda_\Theta \|\Theta\|^2$$

$$L = L_{item} + L_{attr}$$

Multi-task Learning: Optimize for item ranking and attribute ranking simultaneously.

• EAR - Method -- Action stage

Method: Strategy to Ask and Recommend? (Action Stage)

We use **reinforcement learning** to find the best strategy.

- policy gradient method
- simple policy network (2-layer feedforward network)

• **State Vector**

• $s_{entropy}$: The entropy of attribute is important.

• $s_{preference}$: User's preference on each attribute.

• $s_{history}$: Conversation history is important.

• s_{length} : Candidate item list length.

Reward

$r_{success}$: Give the agent a big reward when it successfully recommend!

r_{ask} : Give the agent a small reward when it ask a correct attribute.

r_{quit} : Give the agent a big negative reward when the user quit (the conversation is too long)

Note: 3 of the 4 information come from Recommender Part

Action Space:

$$|\mathcal{P}| + 1$$

$r_{prevent}$: Give each turn a relatively small reward to prevent the conversation goes too long.

• EAR - Method -- Reflection

Method: How to Adapt to User's Online Feedback? (**Reflection stage**)

Solution: We treat the recently rejected 10 items as negative samples to re-train the recommender, to adjust the estimation of user preference.

$$L_{ref} = \sum_{(u,v,v') \in \mathcal{D}_4} -\ln \sigma(\hat{y}(u, v, \mathcal{P}_u) - \hat{y}(u, v', \mathcal{P}_u)) + \lambda_\Theta \|\Theta\|^2$$

Notation	Meaning
\mathcal{V}^t	Recently rejected item set.
$\mathcal{D}_4 := \{(u, v, v') v' \in \mathcal{V}_u^+ \wedge v' \in \mathcal{V}^t\}$	Paired sample for online update.

• EAR - Evaluation

Table 1: Dataset statistics.

Dataset	#users	#items	#interactions	#attributes
Yelp	27,675	70,311	1,368,606	590
LastFM	1,801	7,432	76,693	33

Item Name: “*Small Italy Restaurant*”
Item Attributes: [Pizza, Nightlife, Wine, Jazz]



I'd like some Italian food.

Yes!

Got you, do you like some pizza?

Check, I don't want
“Small Paris”

Yes!

Got you, do you like some nightlife?

Rejected!

Do you want “Small Paris”?

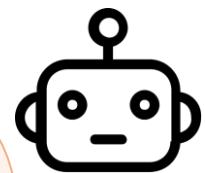
Check, I don't want
“Rock Music”

No!

Got you, do you like some Rock Music?

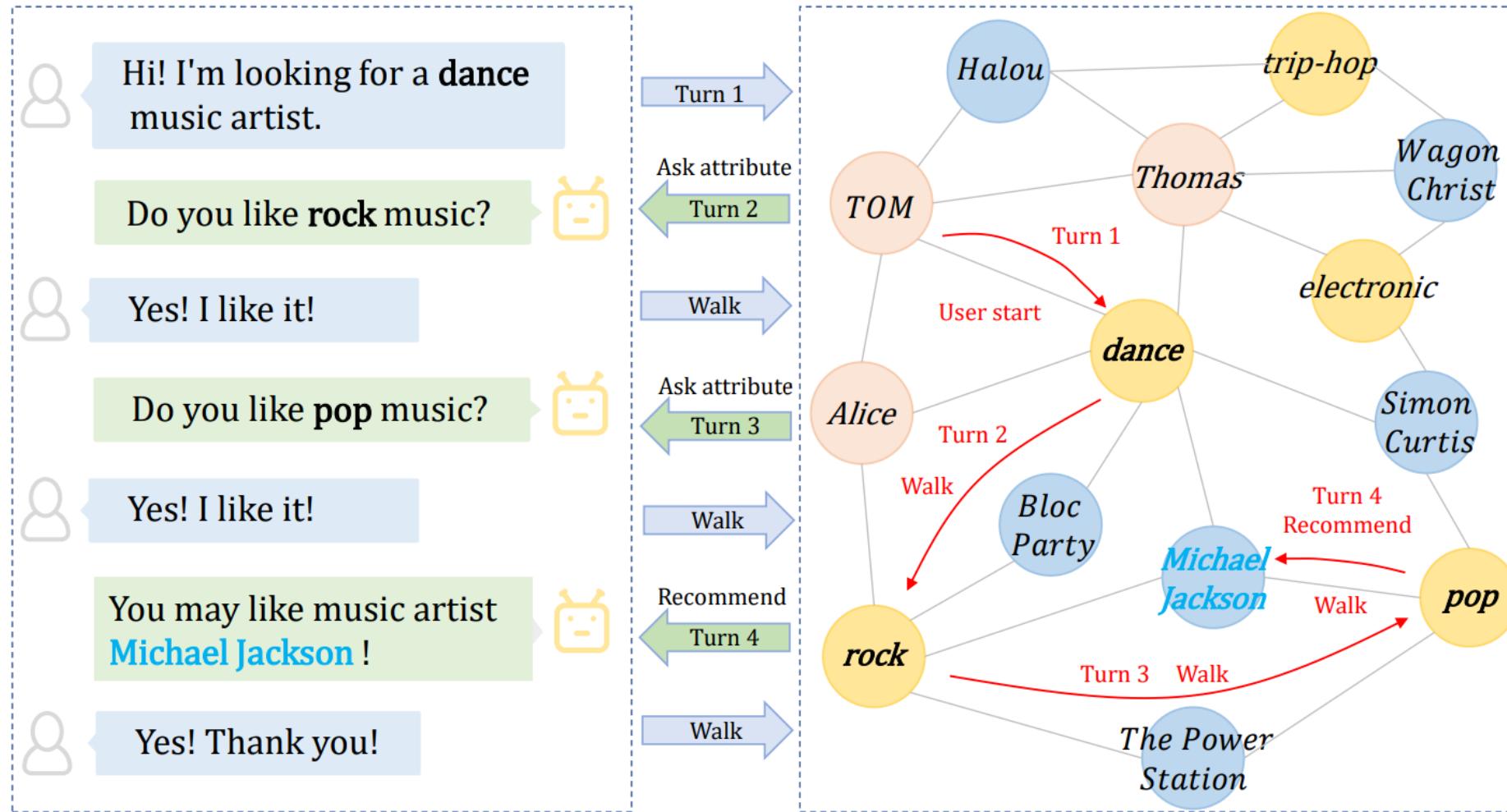
Accepted!

Do you want “Small Italy Restaurant”?



Template-based
utterances

• CPR - Motivation



• CPR - Method

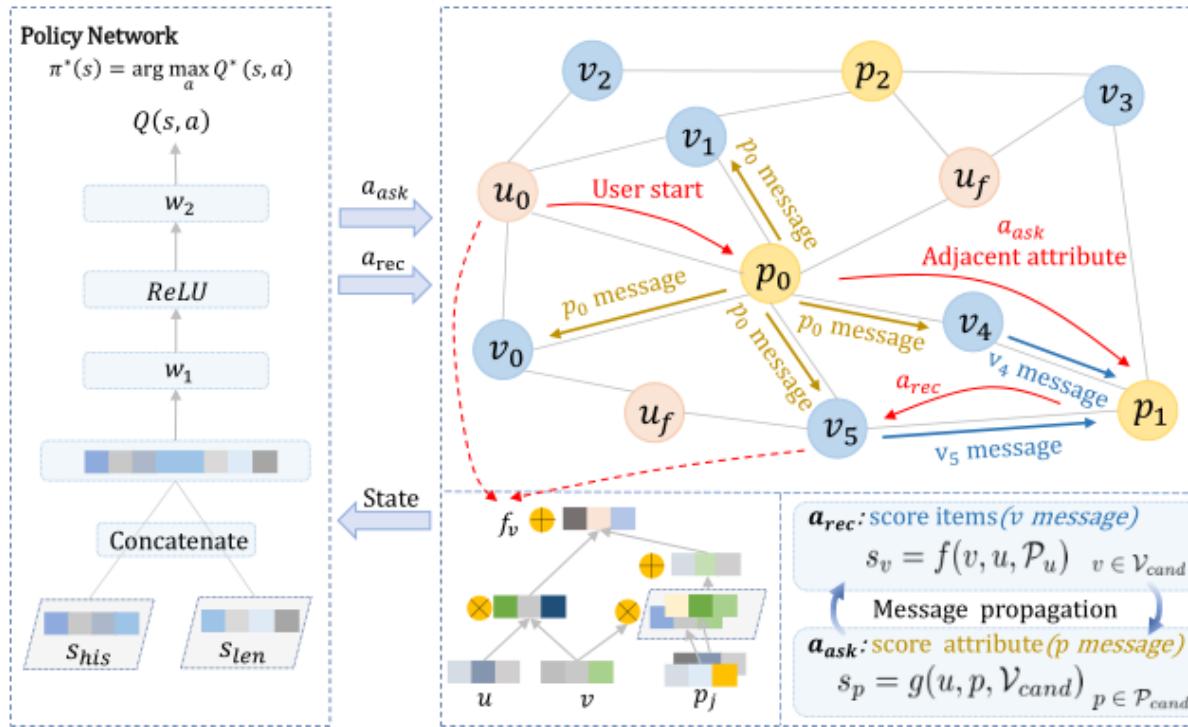


Figure 2: CPR framework overview. It starts from the user u_0 and walks over adjacent attributes, forming a path (the red arrows) and eventually leading to the desired item. The policy network (left side) determines whether to ask an attribute or recommend items in a turn. Two reasoning functions f and g score attributes and items, respectively.

CPR Framework

• Assuming

- Current path $P = p_0, p_1, p_2, \dots, p_t$
- u : user v : item p : attribute
- \mathcal{P}_u : user's preferred attributes
- \mathcal{V}_{cand} : candidate items

• Reasoning

- Score items to recommend (v message):

$$s_v = f(v, u, \mathcal{P}_u) \quad v \in \mathcal{V}_{cand}$$
- Score attribute to ask (p message):

$$s_p = g(u, p, \mathcal{V}_{cand}) \quad p \in \mathcal{P}_{cand}$$

• Consultation

- Policy network (choose to ask or rec)

• Transition

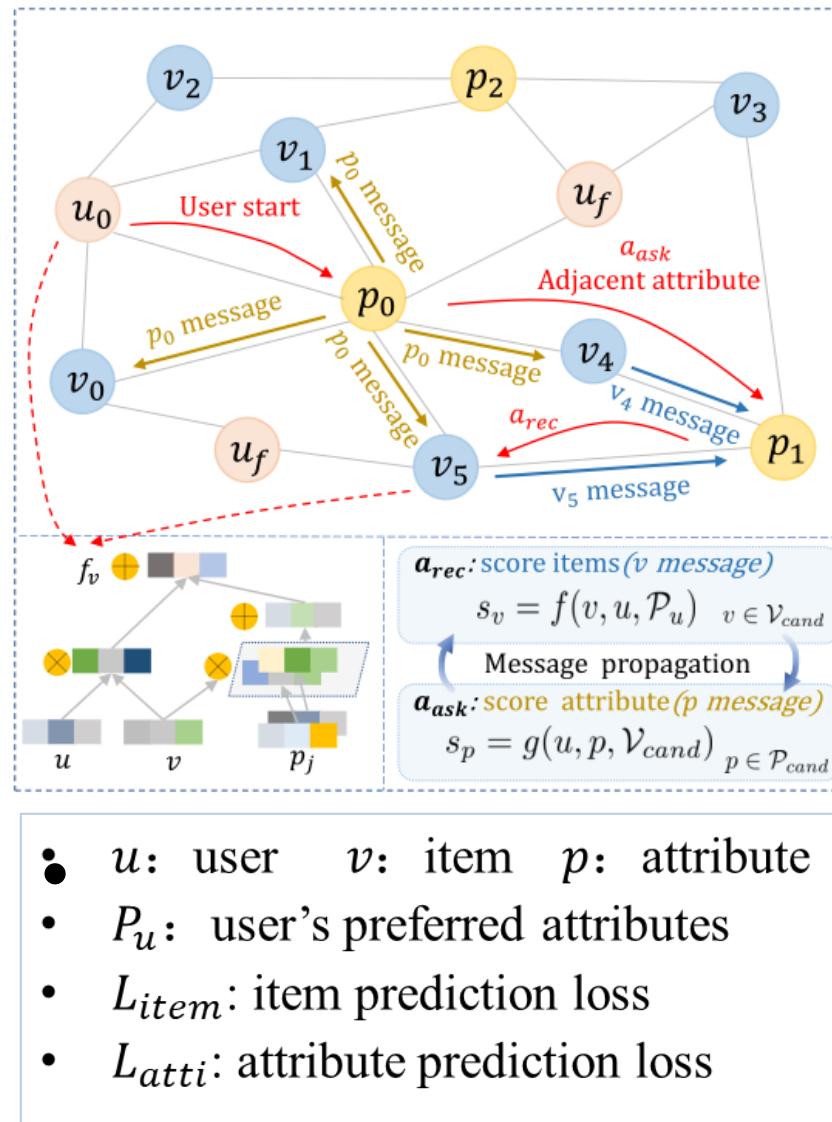
- Extended path

$$P = p_0, p_1, p_2, \dots, p_t, p_{t+1}$$

- Update candidate item /attribute set ($\mathcal{V}_{cand}/\mathcal{P}_{cand}$)

• CPR - Method

An instantiation of CPR
Framework



Message propagation from attributes to items

Factorization Machine in EAR

- Item prediction

$$f(v, u, \mathcal{P}_u) = \mathbf{u}^T \mathbf{v} + \sum_{p \in \mathcal{P}_u} \mathbf{v}^T \mathbf{p},$$

- Optimization:
Bayesian Personalized Ranking

The same with the recommender model in EAR

Message propagation from items to attributes

Information entropy strategy

- Weighted attribute information entropy

$$g(u, p, \mathcal{V}_{cand}) =$$

$$-\text{prob}(p) \cdot \log_2(\text{prob}(p)),$$

$$\text{prob}(p) = \frac{\sum_{v \in \mathcal{V}_{cand} \cap \mathcal{V}_p} \sigma(s_v)}{\sum_{v \in \mathcal{V}_{cand}} \sigma(s_v)}$$

• CPR - Method

Policy Network

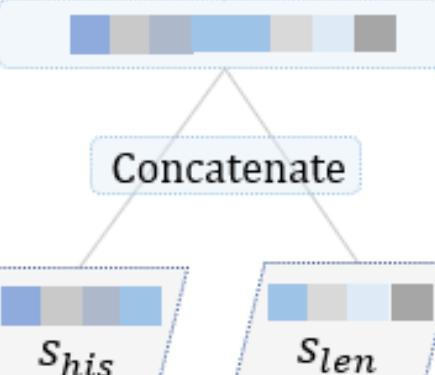
$$\pi^*(s) = \arg \max_a Q^*(s, a)$$

$$Q(s, a)$$

w_2

$ReLU$

w_1



Input

s_{his} : encodes the conversation history

s_{len} : encodes the size of candidate items

Output

$$Q(s, a)$$

$Q(s, a)$: the value of action a in state s

a_{rec} : the action of recommendation

a_{ask} : the action of asking attribute

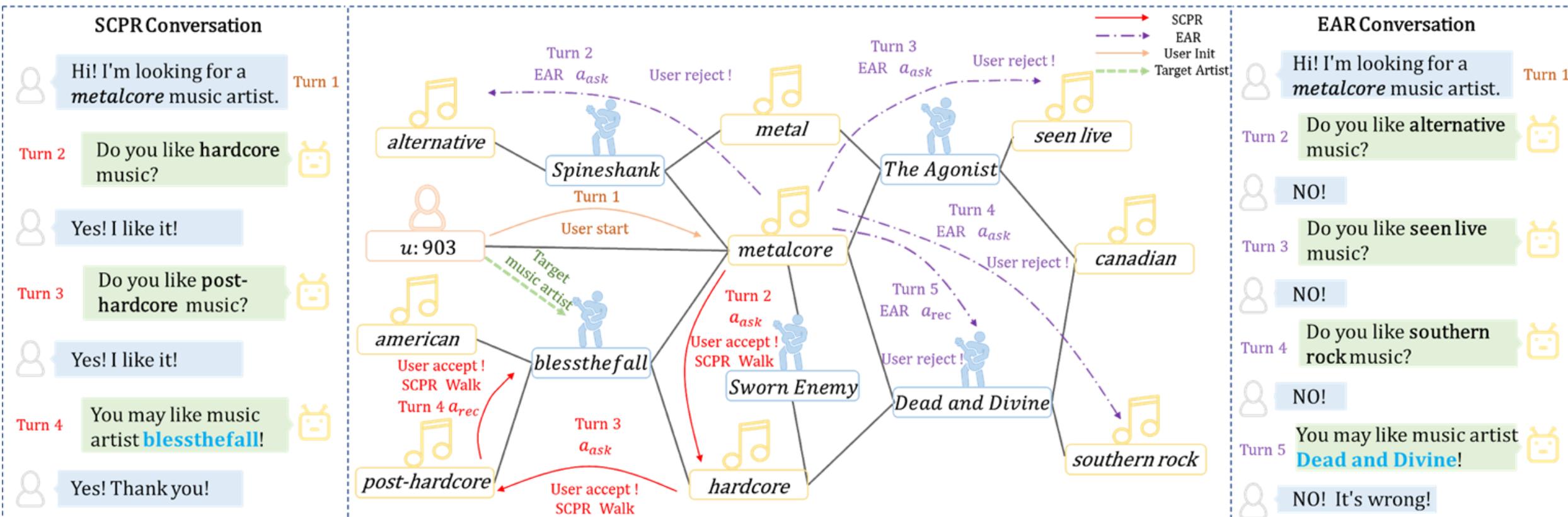
DQN method

$$\text{Policy: } \pi^*(s) = \arg \max_a Q^*(s, a)$$

$$\text{TD loss: } \delta = Q(s, a) - \left(R + \gamma \max_a Q(s', a) \right)$$

• CPR - Evaluation

CPR can make the reasoning process explainable and easy-to-interpret!



Sample conversations generated by SCPR (left) and EAR (right) and their illustrations on the graph (middle).

• Tutorial Outline

- ❑ A Glimpse of Dialogue System
- ❑ Four research directions in conversational recommendation system
 - ❑ Question Driven Approaches
 - ❑ Multi-turn Conversational Recommendation Strategy
 - ❑ **Dialogue Understanding and Generation**
 - ❑ Exploitation-Exploration Trade-offs for Cold Users
- ❑ Summary of Formalizations and Evaluations

• ReDial - Formalization

HUMAN: hello, can i help you find a movie ?
HRED: hi what kind of movies do you like
OURS: hi, how can i help you ?
SEEKER: hello ! i would like to find a suspenseful, but clean family friendly movie .

...
SEEKER: ...
2001 : a space odyssey might be a great option. anything else that you would suggest ?

HUMAN: you can try planet of the apes the older one is quite suspenseful and family friendly .

HRED: i haven't seen that one yet but i heard it was good. i haven't seen that one. have you seen the last house on the left ?

OURS: star wars : the force awakens is also a good one return of the jedi all good movies

SEEKER: those sound good ! i 'm going to look into those movies.

HUMAN: i hope you enjoy, have a nice one

HRED: have you seen foxcatcher ? it 's about a man who has a rich guy.

OURS: i hope i was able to help you find a good movie to watch

SEEKER: thank you for your help ! have a great night ! good bye

Conversational recommendation through natural language (in movie domain)

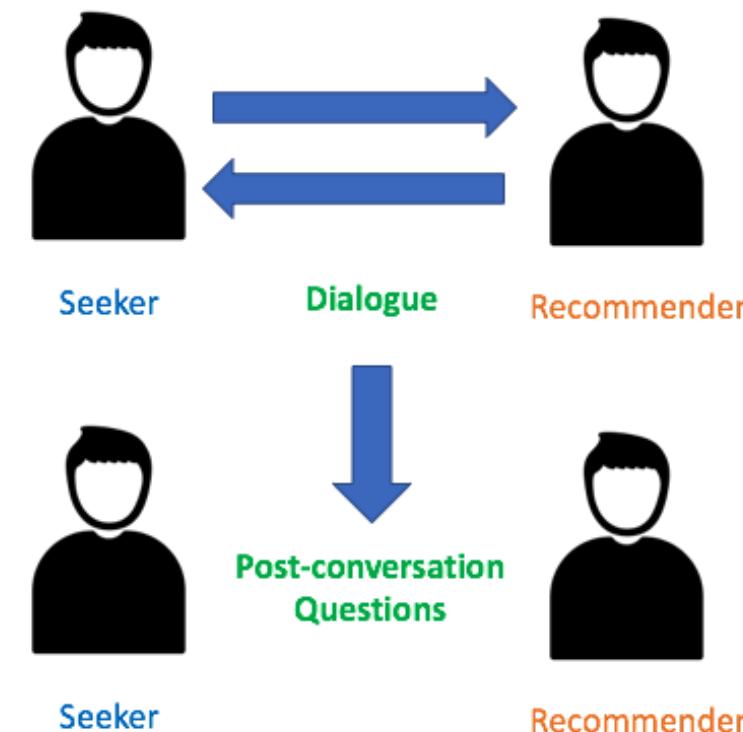
- **Seeker:** explain what kind of movie he/she likes, and asks for movie suggestions
- **Recommender:** understand the seeker's movie tastes, and recommends movies

• ReDial – Formalization -- Dataset Collection

# conversations	10006
# utterances	182150
# users	956
# movie mentions	51699
seeker mentioned	16278
recommender suggested	35421
not seen	16516
seen	31694
did not say	3489
disliked (4.9%)	2556
liked (81%)	41998
did not say (14%)	7145

Data annotation on Amazon Mturk Platform

- 2 turkers: **Seeker** and **recommender** converse with each other.



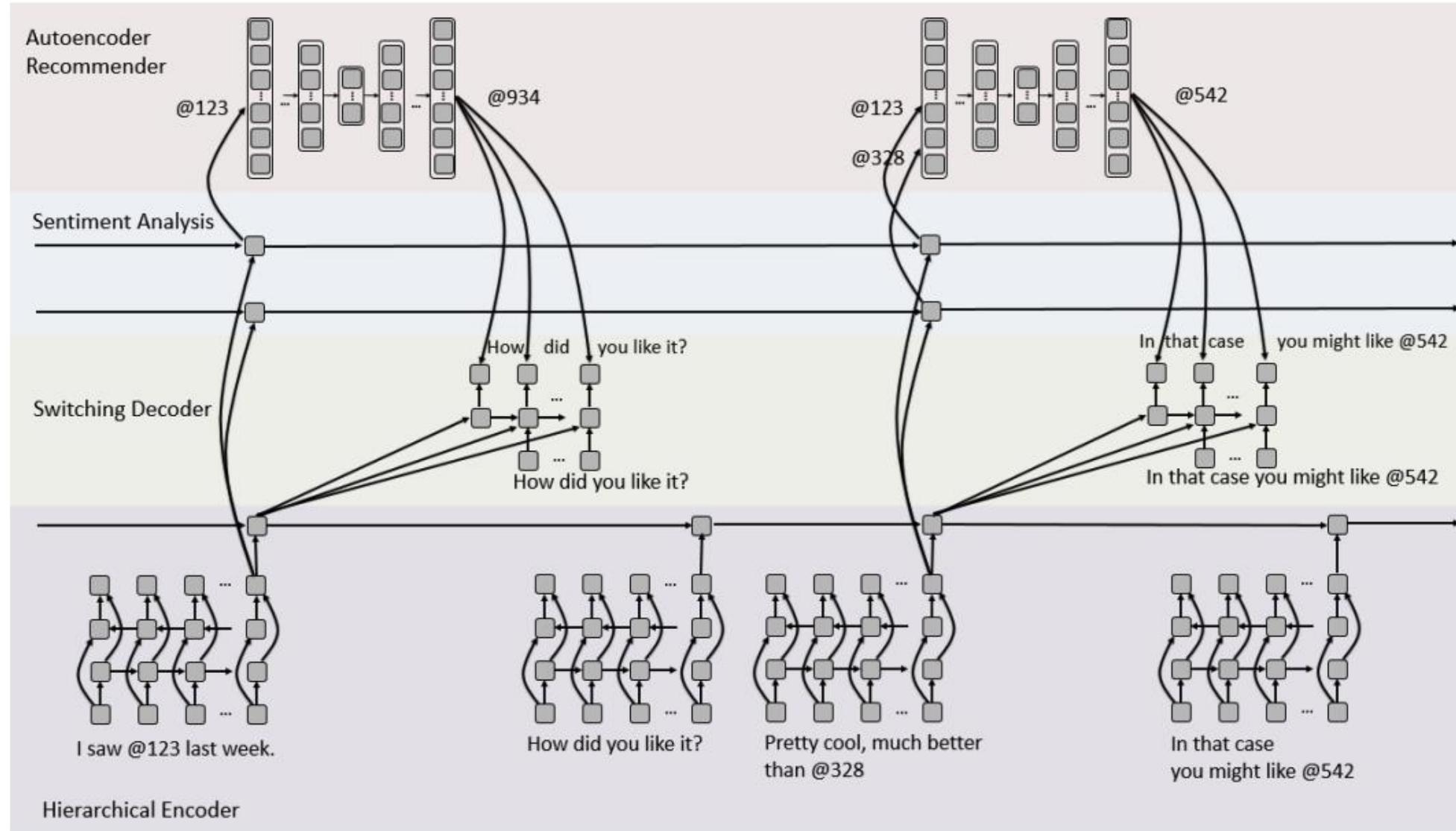
• ReDial – Methods – Overall

3
Recommender

2
Sentiment Analysis

4
Switching Decoder

1
Encoder



Li et al. “Towards Deep Conversational Recommendations” (NIPS’18)

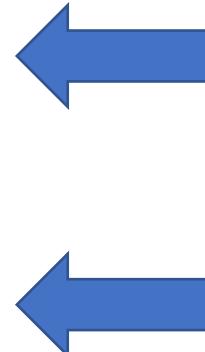
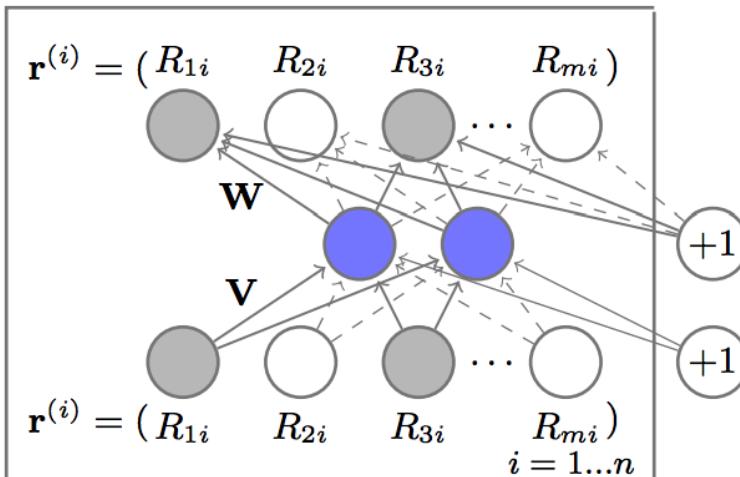
• ReDial – Methods – The Autoencoder Recommender

Notations:

- We have $|M|$ users and $|V'|$ movies.
- User-movie Rating Matrix:
- A user can be represented by

$$\mathbf{R} \in \mathbb{R}^{M \times |V'|} \quad \text{Scale: -1 - 1}$$

$$\mathbf{r}^{(u)} = (\mathbf{R}_{u,1}, \dots, \mathbf{R}_{u,|V'|})$$



Retrieve the full representation from the lower dimension representation

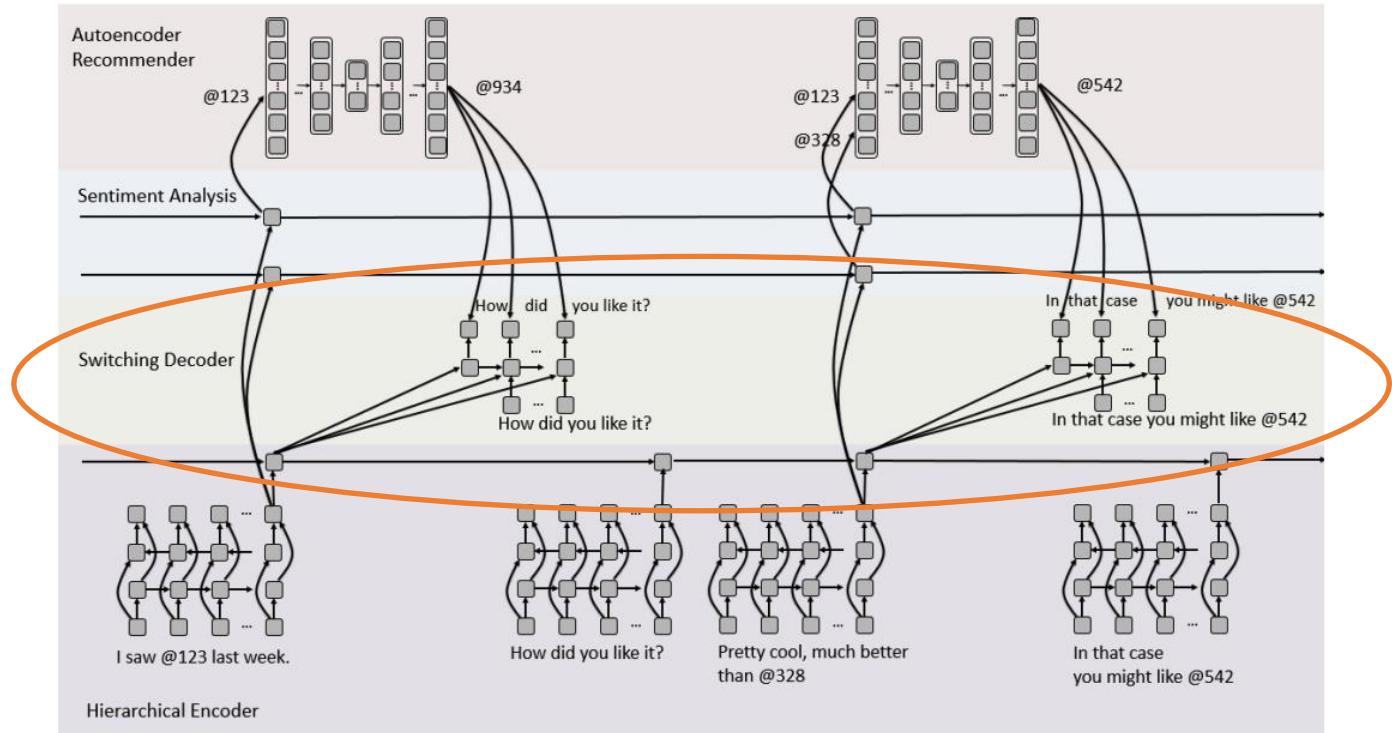
Partially observed user representation fed into a FC layer to lower dimension.

AutoRec: Autoencoders Meet Collaborative Filtering (WWW15)

- Then Loss function:

$$L_{\mathbf{R}}(\theta) = \sum_{u=1}^M \|\mathbf{r}^{(u)} - h(\mathbf{r}^{(u)}; \theta)\|_{\mathcal{O}}^2 + \lambda \|\theta\|^2$$

• ReDial – Methods – Decoder with a Movie Recommendation Switching Mechanism



Responsibility:

- When decoding the next token, decide to mention a movie name, or an ordinary word.

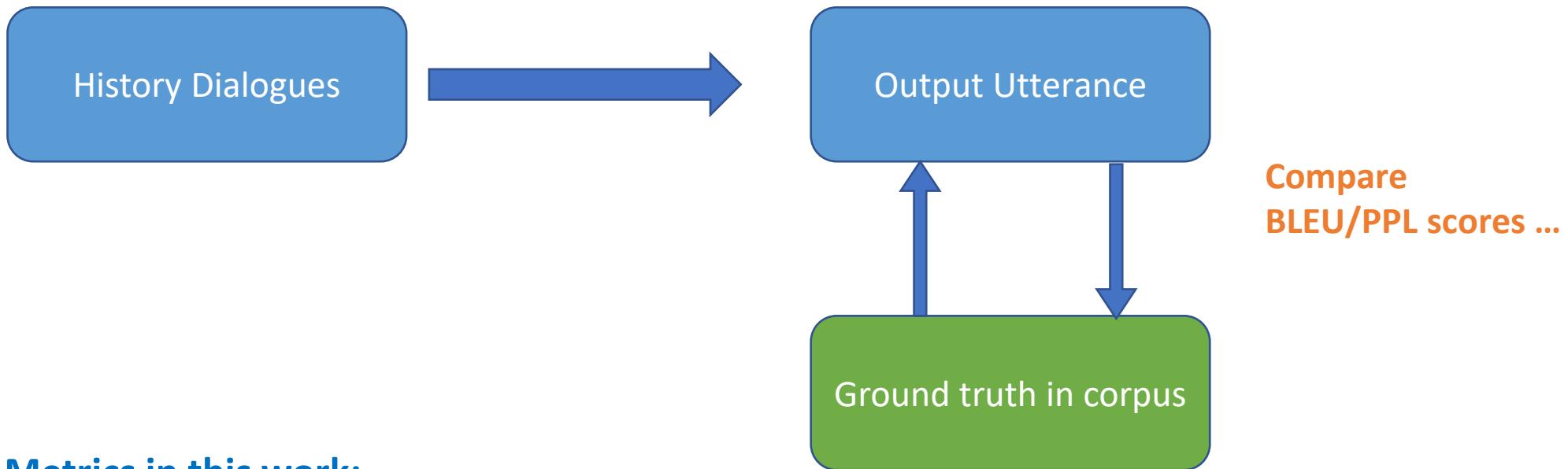
Purpose:

- Such a switching mechanism allows to include an explicit recommendation system in the dialogue agent.

• ReDial – Evaluation – Formalization

Evaluation settings:

Corpus-based evalution. (Similar to the evaluation in dialogue system)



Evaluation Metrics in this work:

- Kappa score: Sentiment analysis subtask
- RMSE score: Recommendation subtask
- Human evaluation: Dialogue generation

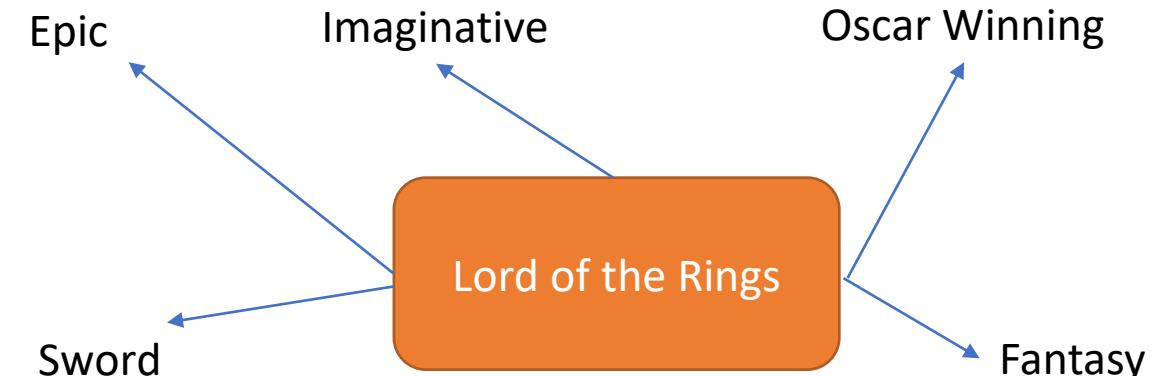
• KBRD – Motivation

The ReDial (NIPS18) paper has two shortage:

- Only mentioned items are used for recommender system.
- Recommender cannot help generate better dialogue.

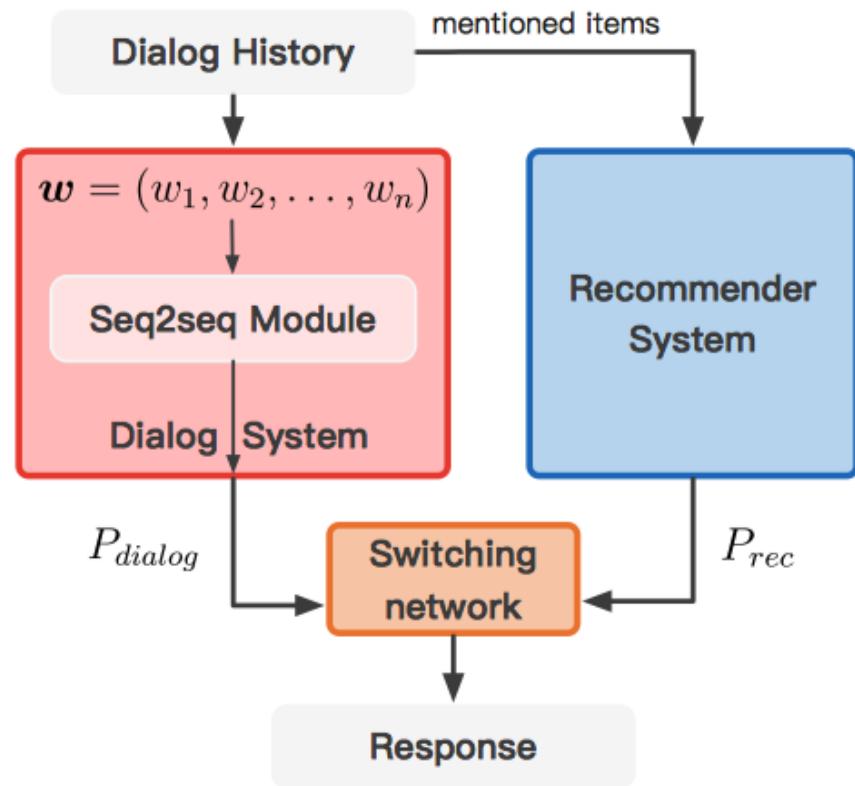


Lord of the Rings is really my all-time-favorite! In fact, I love all **J. R. R. Tolkien**'s work!

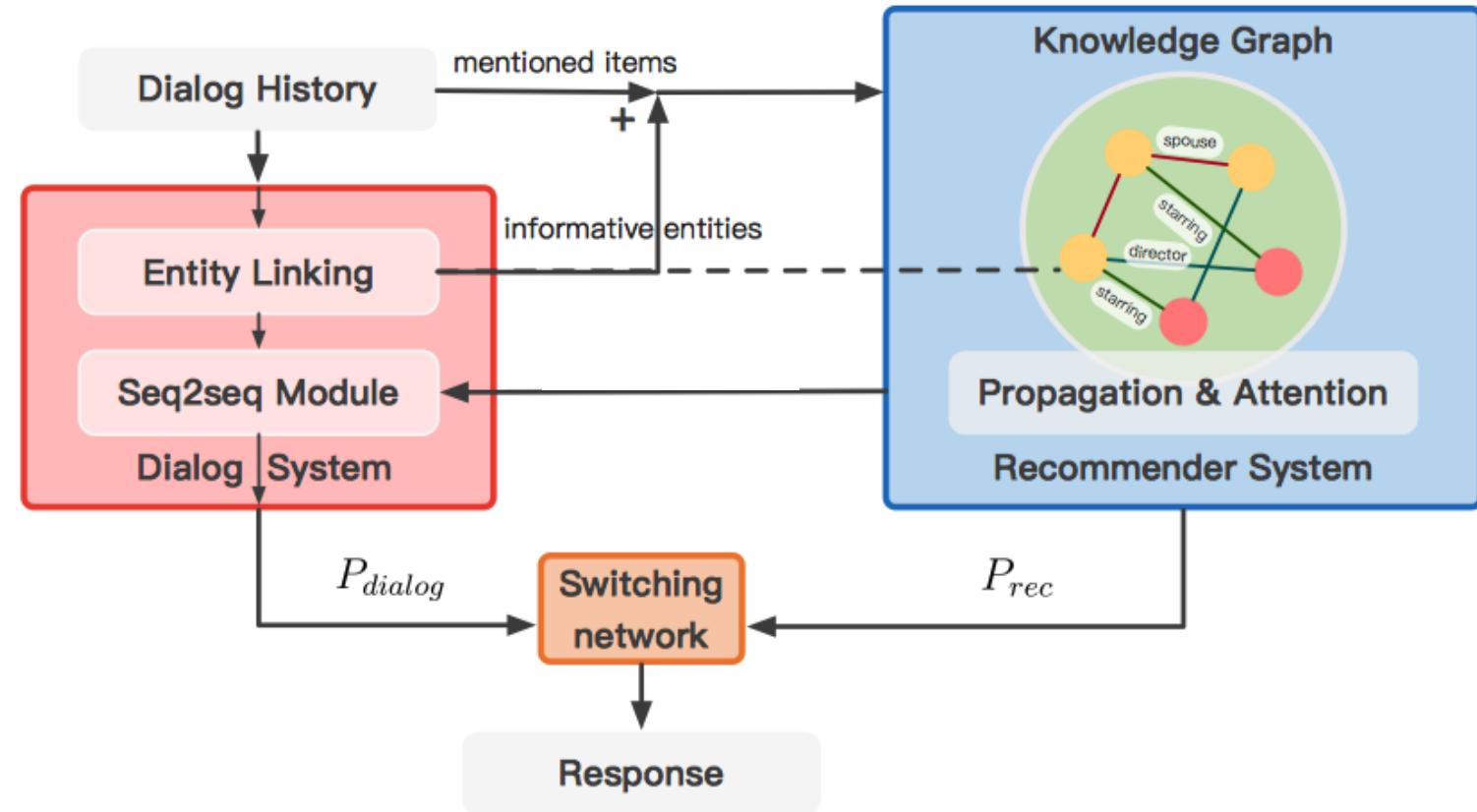


• KBRD – Method – Overall

(a) Baseline



(b) Ours



• KBRD – Experiments – Does Recommendation Help Dialog?

Movie	1	2	3	4	5	6	7	8
Star Wars	space	alien	sci-fi	star	sci	robot	smith	harry
The Shining	creepy	stephen	gory	horror	scary	psychological	haunted	thriller
The Avengers (2012)	marvel	superhero	super	dc	wait	batman	thor	take
Beauty and the Beast	cute	disney	animated	live	music	child	robin	kids

Recommendation-Aware Dialog

$$P_{\text{dialog}} = \text{softmax}(Wo + b)$$

$$b_u = \mathcal{F}(t_u) \quad \text{Vocabulary Bias}$$

$$P_{\text{dialog}} = \text{softmax}(Wo + b + b_u)$$



- We select words with Top 8 vocabulary bias. We can see that these words have strong connection with the movie.

• MGCG – Formalization

Goal planning: QA about <Stolen life>, chitchat about Xun Zhou, recommending the movie <The message>, recommending <Don't cry, Nanking!>

1. **User:** 你知道电影『生死劫』的主演是谁吗? (Who is the star of the movie <stolen life>?)

2. **Bot:** 是周迅哦。 (It is Xun Zhou.)

3. **User:** 是我女神演的呀! (She is my goddess.)

4. **Bot:** 你真有眼光啊, 周迅是中国电视金鹰奖最受欢迎女演员呢。 (You have good taste. Zhou Xun is the most popular actress in the Golden Eagle Award of China TV.)

5. **User:** 是呢, 我特别喜欢她的, 她的演技也很厉害。 (I like her very much. She's very good at acting.)

6. **Bot:** 她还获得过亚洲电影大奖最佳女演员的, 演技是得到过国际认可的。 (She also won the best actress of the Asian Film Awards. Her acting is internationally recognized.)

7. **User:** 嗯嗯, 反正她真的很优秀。 (Anyway, she's really good.)

8. **Bot:** 那你要看看她演的《风声》吗? 该片具有精炼的人物、波谲云诡的情节。很好看哦! (Do you want to see her movie <The message>? It has refined characters and capricious plots.)

9. **User:** 可我现在更喜欢『刘若英』的电影! (But now I prefer Rene Liu's films!)

10. **Bot:** 那刘若英的《南京1937》你可以看看, 电影所展现的是一个导演对战争、对民族、对人性的思考。 (You can see the movie <Don't cry, Nanking>. It shows the director's thinking on war, nation, and human nature.)

11. **User:** 历史片吗? (It is historical film?)

12. **Bot:** 历史战争片。 (Yes. It is historical war film.)

13. **User:** 那我肯定要看看了。 (I will watch it!)

14. **Bot:** 嗯, 你肯定会喜欢的。 (Sure, you will like it!)

15. **User:** 嗯, 谢谢你呀, 下次聊吧! (Thank you! See you later!)

16. **Bot:** 好的, 再见! (OK! Bye!)

QA

Chitchat
about Xun
Zhou

Recommend
<The Message>

Recommend
<Don't Cry,
Nanking>

Recap the settings in NIPS 18:

- **Seeker:** explain what kind of movie he/she likes, and asks for movie suggestions
- **Recommender:** understand the seeker's movie tastes, and recommends movies

The dialogue types are very limited!

In this work, 4 types of dialogues:

- Recommendation
- Chitchat
- QA
- Task

• MGCG – Formalization -- Dataset Collection

Very similar to the dataset collection process as in NIPS 18: Two workers, one for seeker, one for recommender.

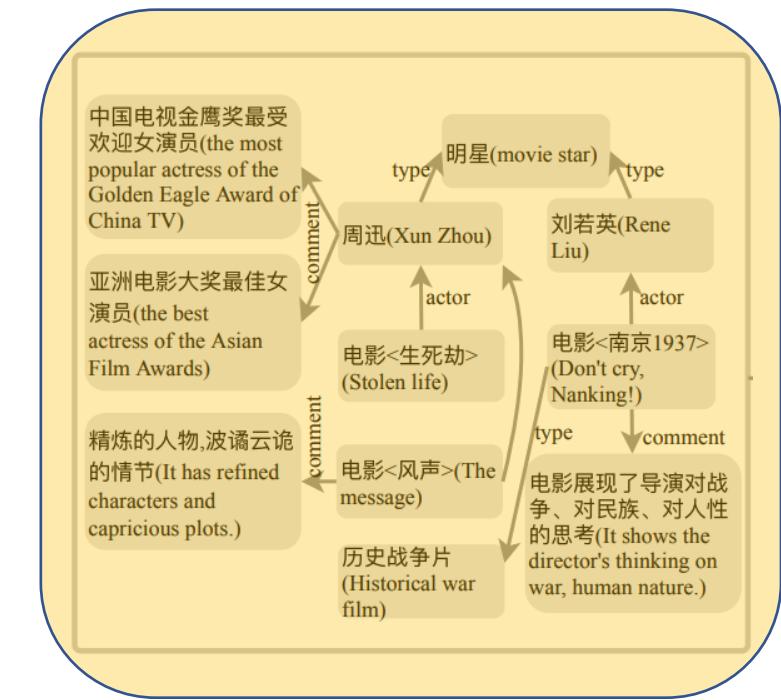
It is further supported by following elements:

Name: 杨凡宇(Fanyu Yang)
Gender: 男(Male)
Age: 20
Domains that the user likes: movie, music
Stars that the user likes: 周迅(Xun Zhou), 刘若英(Rene Liu)
Recommendation accepted: <生死劫>(Stolen Life)
Recommendation rejected: <小王子>(The little prince)

Explicit Seeker Profile
- For the consistency

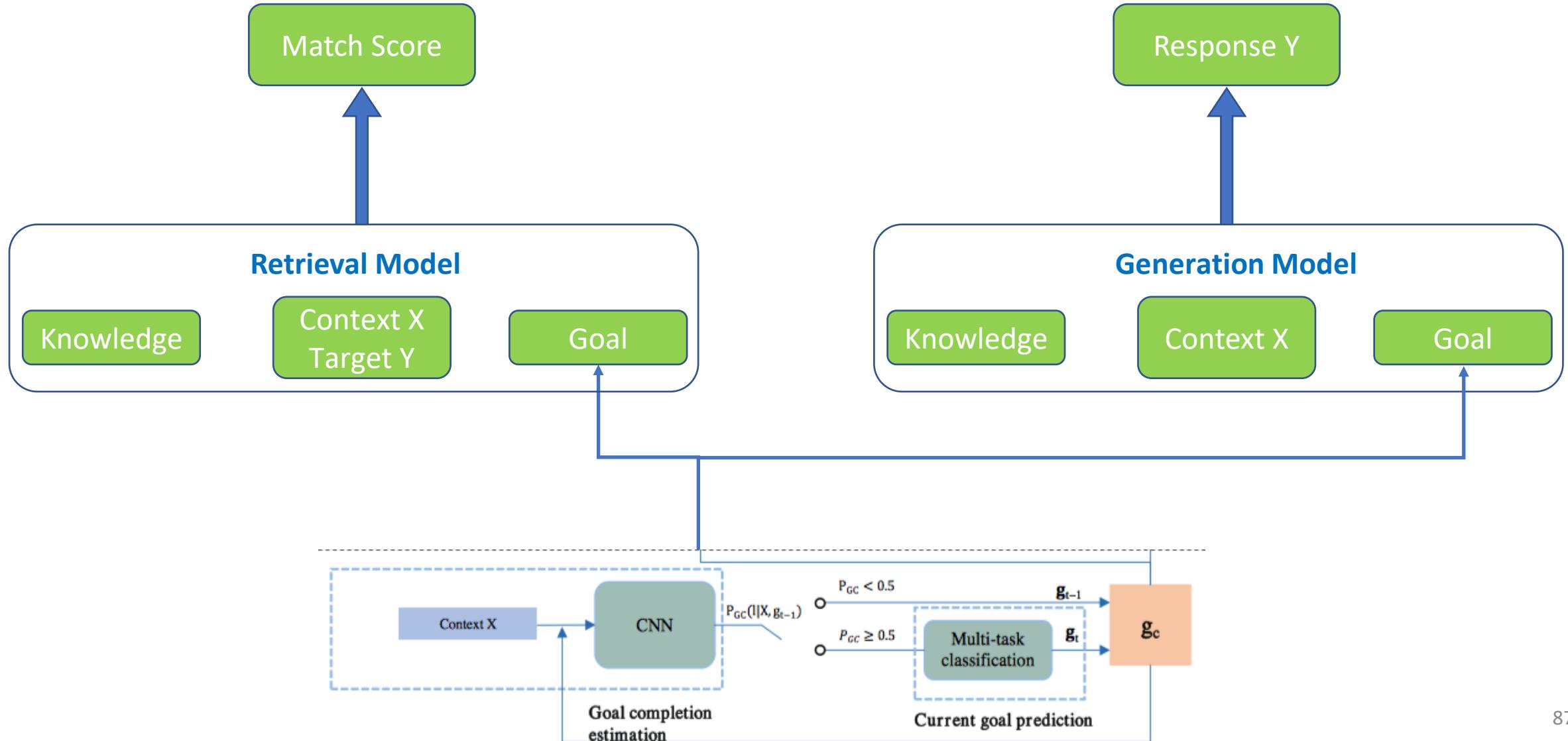
Goals	Goal description
Goal1: QA (dialog type) about the movie <Stolen life> (dialog topic)	The seeker takes the initiative, and asks for the information about the movie <Stolen life>; the recommender replies according to the given knowledge graph; finally the seeker provides feedback.
Goal2: chitchat about the movie star Xun Zhou	The recommender proactively changes the topic to movie star Xun Zhou as a short-term goal, and conducts an in-depth conversation;
Goal3: Recommendation of the movie <The message>	The recommender proactively changes the topic from movie star to related movie <The message>, and recommend it with movie comments, and the seeker changes the topic to Rene Liu's movies;
Goal4: Recommendation of the movie <Don't cry, Nanking!>	The recommender proactively recommends Rene Liu's movie <Don't cry, Nanking!> with movie comments. The seeker tries to ask questions about this movie, and the recommender should reply with related knowledge. Finally the user accepts the recommended movie.

Task Template
- Constrain the complicated task



Knowledge Graph:
- Further assist the workers

• MGCG – Methods

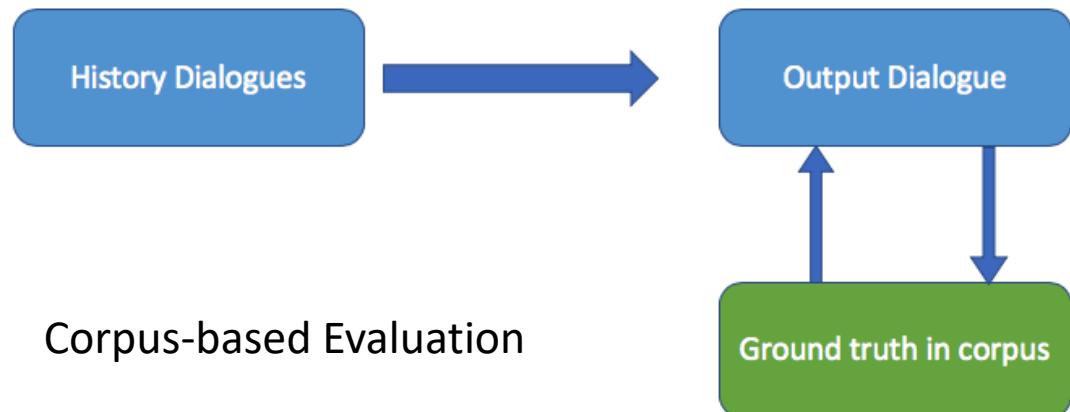


• MGCG – Evaluation – Setting

Evaluation Metrics:

Dialogue generation:

- BLEU – Relevance
- Perplexity – Fluency
- DIST – Diversity
- Hits@1/3 -- Retrieval model (1 ground truth, 9 randomly sampled.)



Corpus-based Evaluation

Human Evaluation:

- Turn level: fluency, appropriateness, informativeness, and proactivity.
- Dialogue level: Goal success rate and Coherence

• KMD – Motivation and Formalization



Motivation:

Existing dialogue systems only utilize textual information, which is not enough for full understanding of the dialogue.

- What is “these”?
- What is “it”?

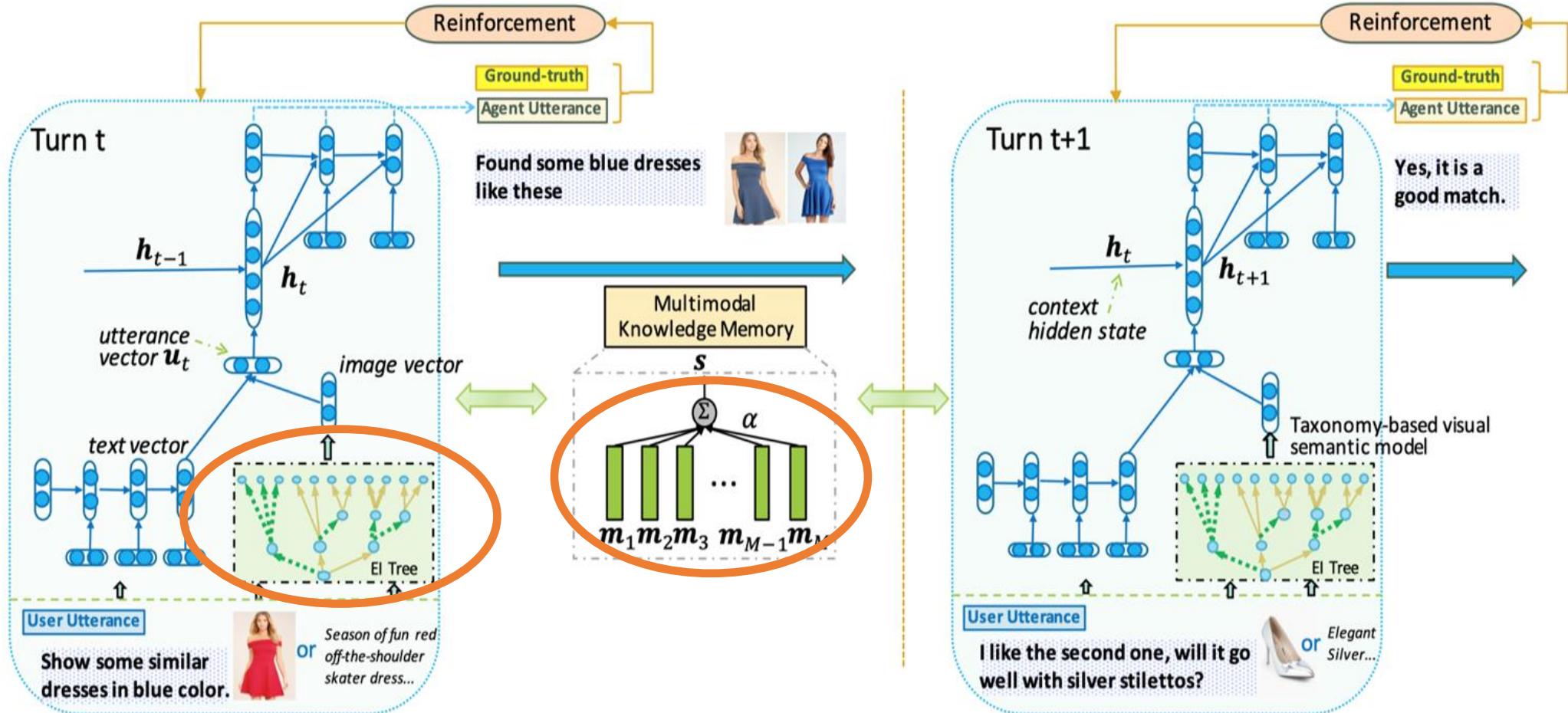
Background: Fashion Match!

User utterance u_1, u_2, \dots, u_t $\xrightarrow{\quad} \hat{u}_t$

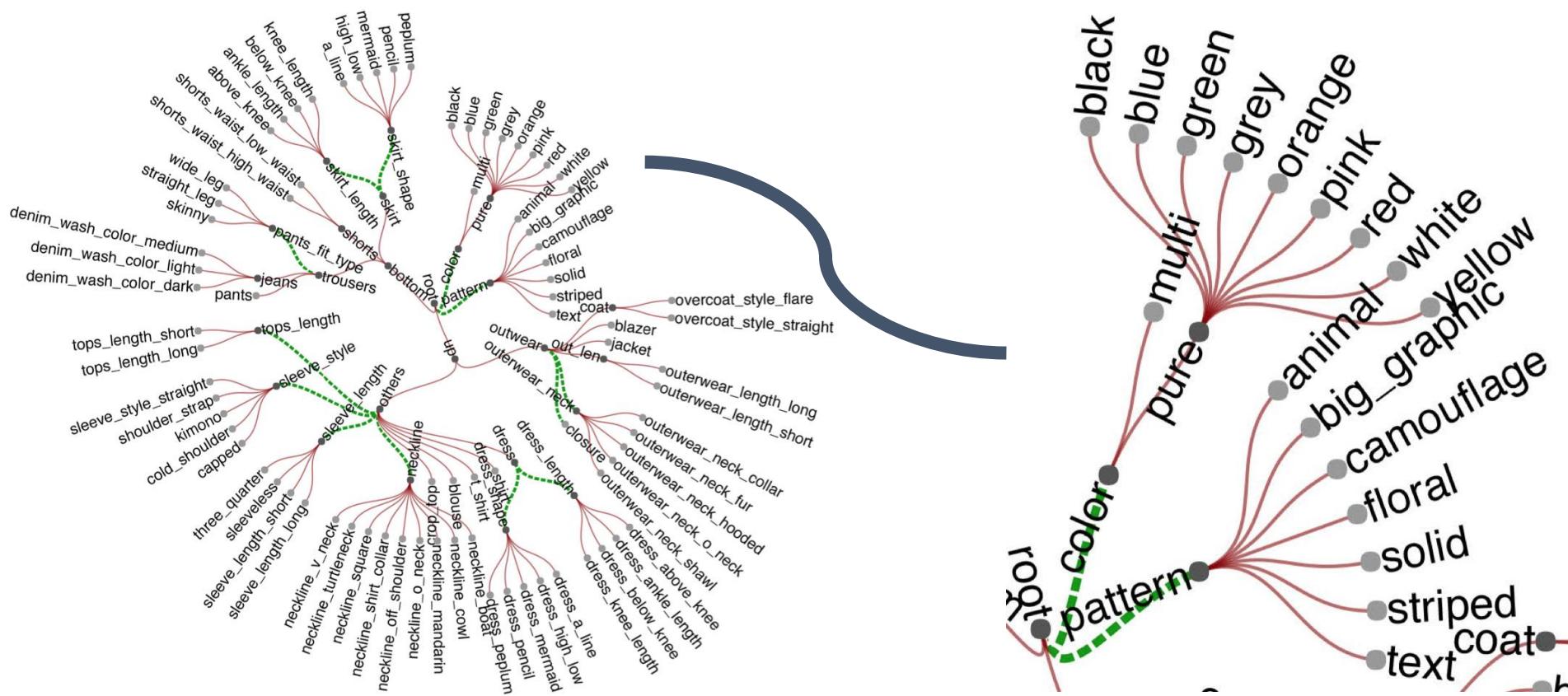
Agent utterance $\hat{u}_1, \hat{u}_2, \dots, \hat{u}_{t-1}$

u be both Text and Image modality

• KMD – Method – Overview



• KMD – Method – Exclusive & Inclusive Tree (EI Tree)



Instead of CNN to capture image feature, they used taxonomy-based feature. They argued that CNN only captures generic features, but they want to capture the rich domain knowledge in specific domain.

• KMD – Method – EI Tree

Encode text features

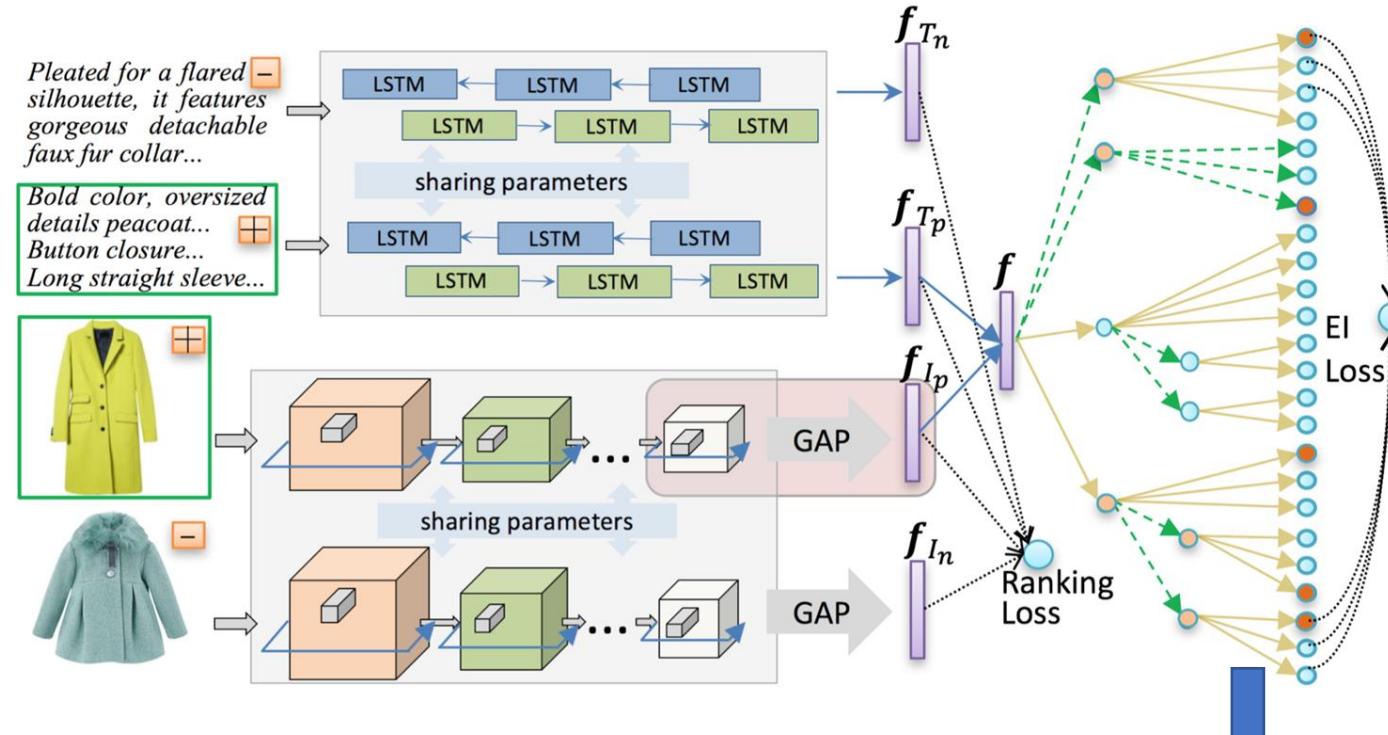
Pleated for a flared –
silhouette, it features
gorgeous detachable
faux fur collar...

Bold color, oversized
details peacoat...
Button closure...
Long straight sleeve...

Encode image features



A sequence of steps along the path.



$$p(c_n | c_0 \rightarrow c_n, \mathbf{f}, \mathbf{W}_{EI}) = \\ p(c_1 | c_0, \mathbf{f}, \mathbf{W}_{EI}) \cdot p(c_2 | c_1, \mathbf{f}, \mathbf{W}_{EI}) \cdots p(c_n | c_{n-1}, \mathbf{f}, \mathbf{W}_{EI}),$$

Optimization:

- **EI Loss:** Compare the predicted leaf node against ground truth, and optimize the cross entropy loss.
- **Pairwise ranking loss** is used to regularize the model to match text and image feature.

• KMD – Method – Incorporation of Domain Knowledge

Fashion Tips: if the user asks for advice about matching tips of **NUS hoodie**, the matching candidates such as the **Livi's jeans** might not co-occur with it in the whole training corpus or conversation history.



- KMD – Method – Incorporation of Domain Knowledge

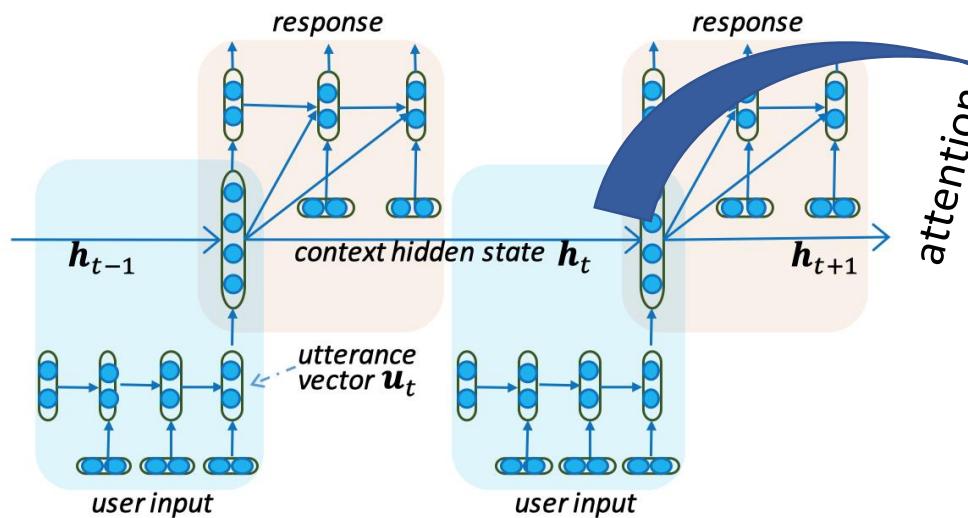
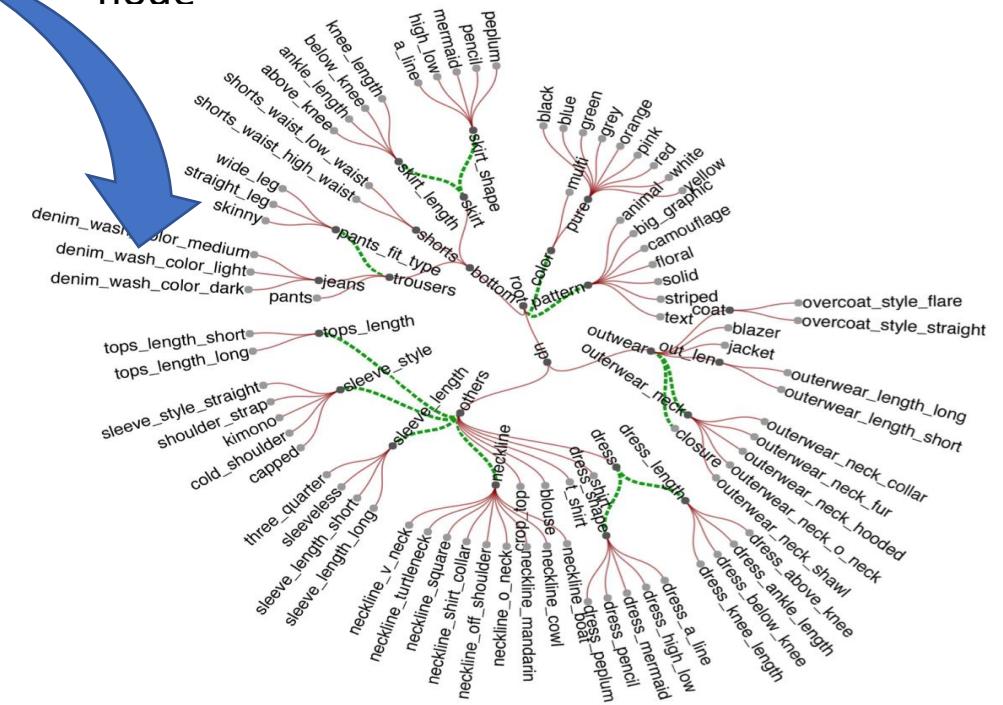


Figure 5: An illustration of text-based HRED backbone

They incorporated knowledge into HRED model (hierarchical recurrent encoder-decoder)

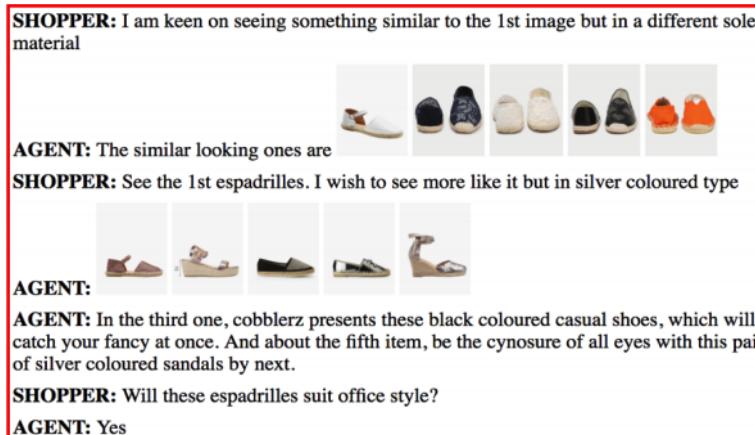
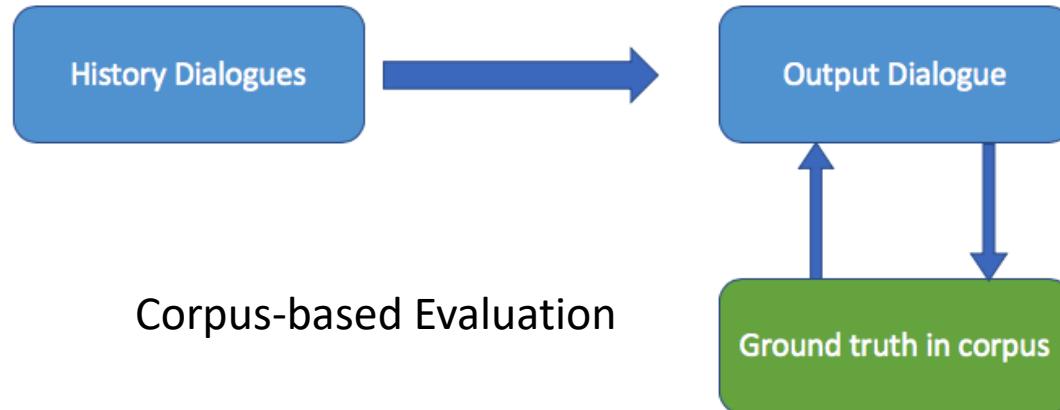
Each EI tree leaf gets a memory vector:
the averaging of the image representation corresponds to the leaf node



$$\mathbf{h}'_t = \mathbf{h}_t + \mathbf{s}.$$

S is the weighted sum
of the memory vector

• KMD – Evaluation – Formalization



Evaluation Metrics:

Text generation:

- BLEU Score
- Diversity (unigram)

Image response generation:

- Recall @ K

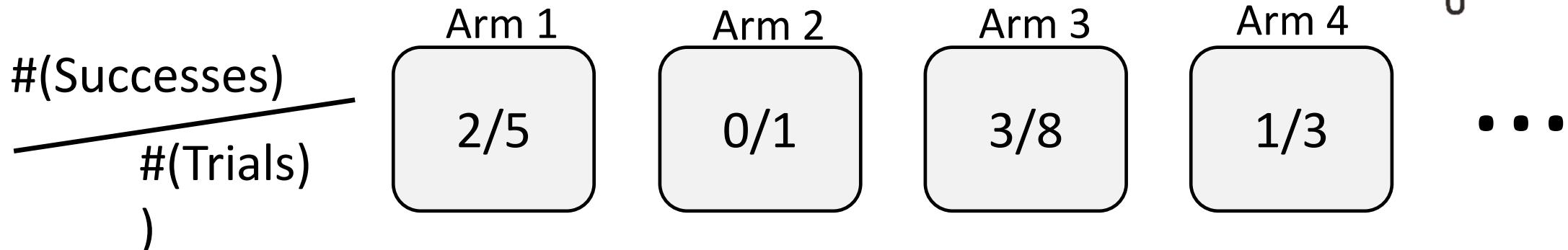
• Tutorial Outline

- ❑ A Glimpse of Dialogue System
- ❑ Four research directions in conversational recommendation system
 - ❑ Question Driven Approaches
 - ❑ Multi-turn Conversational Recommendation Strategy
 - ❑ Dialogue Understanding and Generation
 - ❑ **Exploitation-Exploration Trade-offs for Cold Users**
- ❑ Summary of Formalizations and Evaluations

• Bandit algorithms for Exploitation-Exploration trade-off



Multi-armed bandit example: which arm to select next? 



Common intuitive ideas:

- **Greedy:** trivial exploit-only strategy
- **Random:** trivial explore-only strategy
- **Epsilon-Greedy:** combining Greedy and Random.
- **Max-Variance:** only exploring w.r.t. uncertainty.

• Upper Confidence Bounds (UCB) - Method

Arm selection strategy:

$$\hat{a} = \arg \max_a \hat{Q}(a) + \Delta(a)$$

Exploitation **Exploration**

$\hat{Q}(a) = \frac{1}{N} \sum_{t=1}^{N_a} r_{t,a}$: The estimated mean of reward of arm a .
 $\Delta(a)$: The uncertainty of $\hat{Q}(a)$.

Estimating rewards by averaging the observed rewards:

$$\begin{array}{c} \#(\text{Successes}) \\ \diagdown \\ \#(\text{Trials}) \\) \end{array}$$

Arm 1
 $\hat{Q}(a) = 2/5$

Arm 2
 $\hat{Q}(a) = 0/1$

Arm 3
 $\hat{Q}(a) = 3/8$

Arm 4
 $\hat{Q}(a) = 1/3$

...

$$\hat{a} = \arg \max_a \hat{Q}(a) + \alpha \sqrt{\frac{\log T}{t_a}}$$

- A Contextual-Bandit Approach with Linear Reward (LinUCB) - Method

Estimating reward by introducing the

feature vector $\mathbf{X}_{t,a}^T$: $\hat{Q} = \mathbf{X}_{t,a}^T \boldsymbol{\theta}_a$

#(Successes)	Arm 1 $\mathbf{X}_{t,a}^T \boldsymbol{\theta}_a =$ 2/5	Arm 2 $\mathbf{X}_{t,a}^T \boldsymbol{\theta}_a =$ 0/1	Arm 3 $\mathbf{X}_{t,a}^T \boldsymbol{\theta}_a =$ 3/8	Arm 4 $\mathbf{X}_{t,a}^T \boldsymbol{\theta}_a =$ 1/3	...
#(Trials))					

The arm selection strategy is:

Exploitation
Exploration

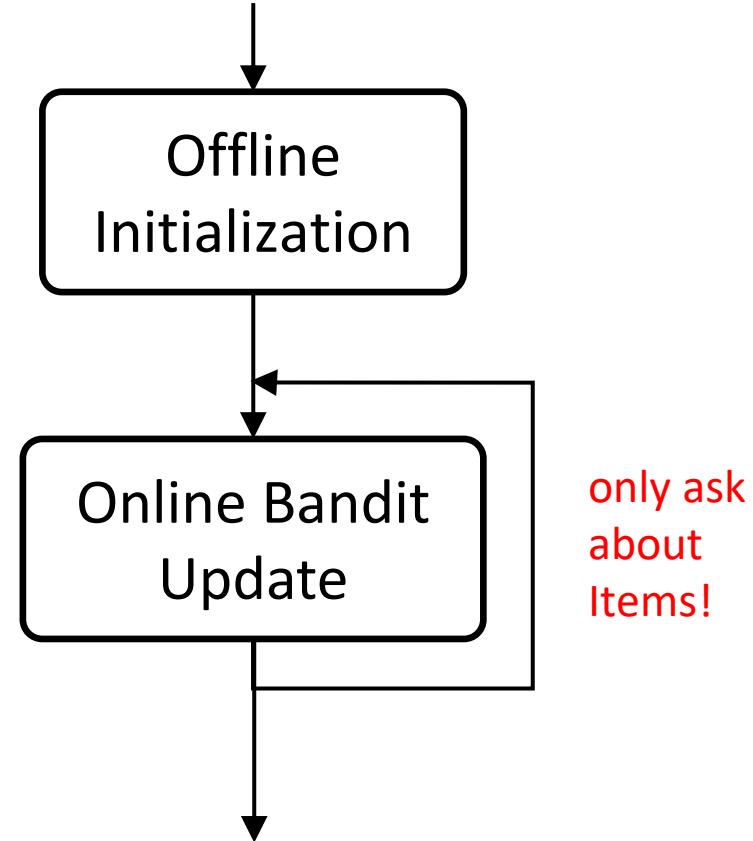
$$a_t \stackrel{\text{def}}{=} \arg \max_a \left(\boxed{\mathbf{X}_{t,a}^T \boldsymbol{\theta}_a} + \alpha \sqrt{\boxed{\mathbf{X}_{t,a}^T \mathbf{A}_a^{-1} \mathbf{X}_a}} \right)$$

where $\mathbf{A}_a \stackrel{\text{def}}{=} \mathbf{D}_a^T \mathbf{D}_a + \mathbf{I}_a$
 $\alpha = 1 + \sqrt{\ln(2/\delta)/2}$

• Bandit algorithm in Conversational Recommendation System - Formalization

Setting:

- For cold start users, the user embedding is initialized as the average embedding of existing users.
- Asking only whether a user likes items (no attributes questions).
- The model updates its parameters at each turn.



• Bandit algorithm in Conversational Recommendation System - Method

Method:

Traditional recommendation model

Absolute Model. First, let us assume that we have observed tuples of the form (user i , item j , 1/0).⁴ The model estimates the *affinity* of user i to item j based on the biases and traits. The generative procedure is:

1. User i has traits $\mathbf{u}_i \sim \mathcal{N}(\mathbf{0}, \sigma_1^2 \mathbf{I})$, bias $\alpha_i \sim \mathcal{N}(0, \sigma_2^2)$.
2. Item j has traits $\mathbf{v}_j \sim \mathcal{N}(\mathbf{0}, \sigma_1^2 \mathbf{I})$, bias $\beta_j \sim \mathcal{N}(0, \sigma_2^2)$.
3. (a) The (unobserved) affinity is

$$y_{ij} = \alpha_i + \beta_j + \mathbf{u}_i^T \mathbf{v}_j. \quad (1)$$

Observations are modeled as the noisy estimate $\hat{y}_{ij} \sim \mathcal{N}(y_{ij}, \epsilon_{ij})$, where ϵ_{ij} models the affinity variance, accounting for noise in user preferences. This yields an observation of whether the user likes an item (\hat{r}_{ij}):

$$\hat{r}_{ij} = \mathbf{1}[\hat{y}_{ij} > 0]. \quad (2)$$

Traditional MF-based recommendation model

+

bandit model

- Terminology :
- trait=embedding**

Greedy: $j^* = \arg \max_j y_{ij}$

A trivial *exploit*-only strategy: Select the item with highest estimated affinity mean.

Random: $j^* = \text{random}(1, N)$

A trivial *explore*-only strategy.

Maximum Variance (MV): $j^* = \arg \max_j \epsilon_{ij}$

A *explore*-only strategy, variance reduction strategy: Select the item with the highest noisy affinity variance.

Maximum Item Trait (MaxT): $j^* = \arg \max_j \|\mathbf{v}_j\|_2$

Select the item whose trait vector \mathbf{v}_j contains the most information, namely has highest L2 norm $\|\mathbf{v}_j\|_2 = \sqrt{v_{j1}^2 + v_{j2}^2 + \dots + v_{jd}^2}$.

Minimum Item Trait (MinT): $j^* = \arg \min_j \|\mathbf{v}_j\|_2$

Select the item with trait vector with least information.

Upper Confidence (UCB): $j^* = \arg \max_j y_{ij} + \epsilon_{ij}$

Based on UCB1 [3]: Pick the item with the highest upper confidence bound, namely mean plus variance (95% CI)

Thompson Sampling (TS) [5]: $j^* = \arg \max_j \hat{y}_{ij}$

For each item, sample the noisy affinity from the posterior. Select item with the maximum sampled value.

Common bandit strategies

- Bandit algorithm in Conversational Recommendation System - Evaluation

Setting: Offline initialization + Online updating

- Online stage: Ask 15 questions of 10 items. Each question is followed by a recommendation.
- Metric: Average precision $AP@10$, which is a widely used recommendation metric.

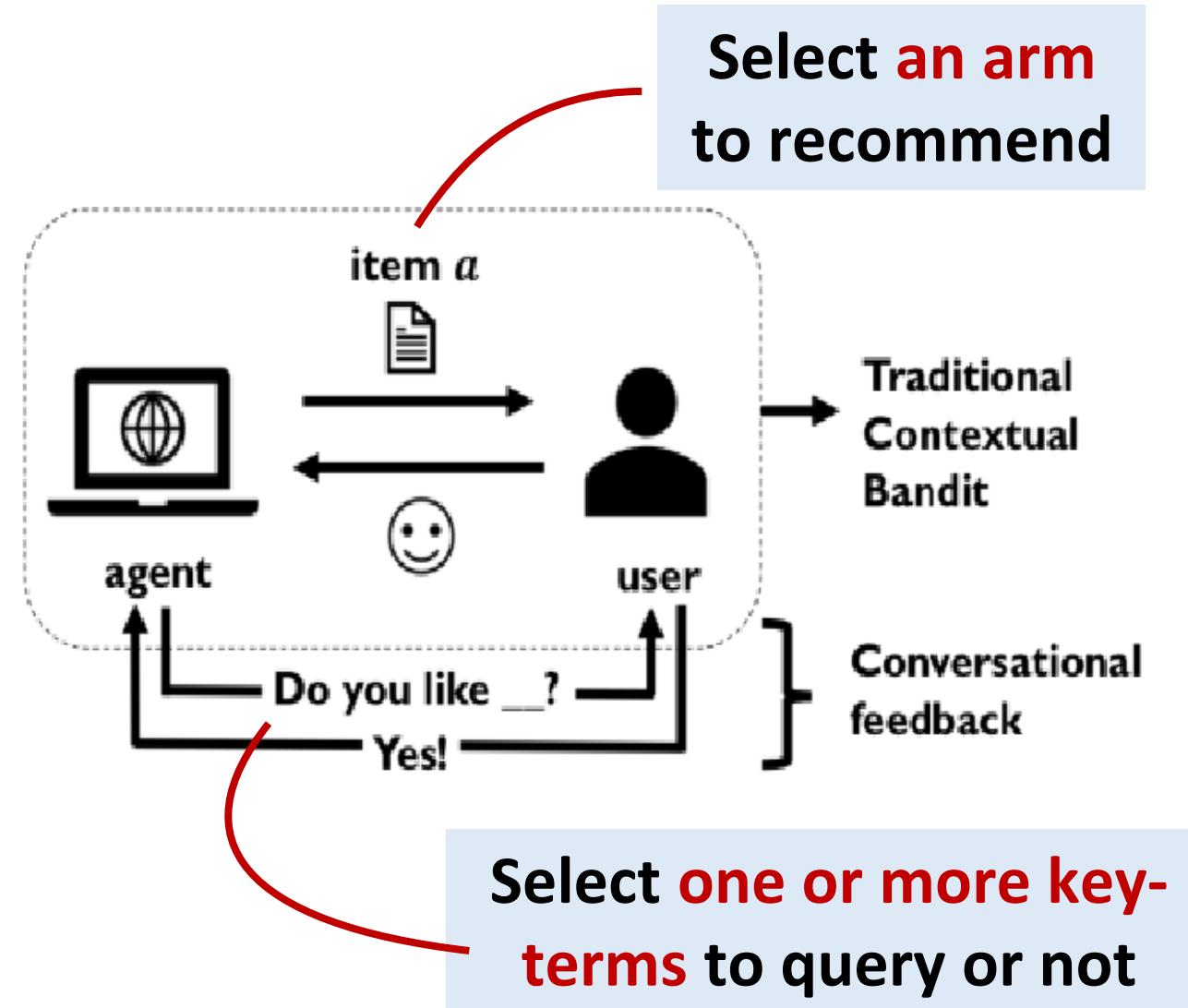
Real data: collected from restaurant searching logs

- Offline learning on collected **3549 users 289 restaurants, and 9330 positive observations.**
- Recruit **28 users** to rate on the **selected 10 restaurants**.
- Online cold-start user study: **each one of the 28 users rates 10 carefully selected restaurants, based on which his/her preference u_i is inferred. Then, run 50 times:**
 1. Sample a user i .
 2. Sample $\hat{u}_i \sim u_i$.
 3. Use \hat{u}_i to simulate reward of each restaurant.

• Conversational UCB algorithm(ConUCB) - Formalization

Setting:

- Asking questions about not only the **bandit arms (items)**, but also the **key-terms (categories, topics)**.
- **One key-term is related to a subset of arms. Users' preference on key-terms can propagate to arms.**
- **Each arm has its own features.**



• ConUCB - Method -- Overview

Select attributes (key-terms) to query

Select an item (arm) to recommend

Algorithm 1: General algorithm of ConUCB

Input: arms \mathcal{A} , key-terms \mathcal{K} , graph $(\mathcal{A}, \mathcal{K}, W)$, $b(t)$.

- 1 **for** $t = 1, \dots, T$ **do**
- 2 observe contextual vector $\mathbf{x}_{a,t}$ of each arm $a \in \mathcal{A}_t$;
- 3 If conversation is allowed at round t , i.e., $q(t) = 1$, select key-terms to conduct conversations and receive conversational feedbacks $\{\tilde{r}_{k,t}\}$;
- 4 select an arm $a_t = \arg \max_{a \in \mathcal{A}_t} \tilde{R}_{a,t} + C_{a,t}$;
- 5 receive a reward $r_{a_t,t}$;
- 6 update model ;

Exploitation Exploration

• ConUCB - Method

When to query the key-terms:

- Define a function $b(t)$, which determines:
 - (1) whether to converse at round t .
 - (2) the number of conversations until round t .

- Consider the function $q(t)$:

$$q(t) = \begin{cases} 1, & b(t) - b(t-1) > 0, \\ 0, & \text{otherwise.} \end{cases}$$

- If $q(t) = 1$, query about $b(t) - b(t-1)$ key-terms;
- If $q(t) = 0$, does not query about a key-term;
- For users' experience, key-term-level conversations should be less frequent than arm-level interactions, i.e., $b(t) \leq t, \forall t$.

Examples:

- 1) The agent makes k conversations in every m rounds.

$$b(t) = k \left\lfloor \frac{t}{m} \right\rfloor, m \geq 1, k \geq 1.$$

- 1) The agent makes a conversation with a frequency represented by the logarithmic function of t .

$$b(t) = \lfloor \log(t) \rfloor$$

- 1) There is no conversation between the agent and the user.

$$b(t) \equiv 0$$

• ConUCB - Method

The core strategy to select arms and key-terms:

- Selecting the arm with the largest upper confidence bound derived from both arm-level and key-term-level feedback, and receives a reward.

User preference computed on key-term-level rewards

$$\tilde{\theta}_t = \arg \min_{\tilde{\theta}} \sum_{\tau=1}^t \sum_{k \in \mathcal{K}_\tau} \left(\frac{\sum_{a \in \mathcal{A}} w_{a,k} \tilde{\theta}^T \mathbf{x}_{a,\tau}}{\sum_{a \in \mathcal{A}} w_{a,k}} - \tilde{r}_{k,\tau} \right)^2 + \tilde{\lambda} \|\tilde{\theta}\|_2^2,$$

User preference computed on arm-level rewards

$$\theta_t = \arg \min_{\theta} \lambda \sum_{\tau=1}^{t-1} (\theta^T \mathbf{x}_{a_\tau, \tau} - r_{a_\tau, \tau})^2 + (1-\lambda) \|\theta - \tilde{\theta}_t\|_2^2.$$

Constrain θ to
be close to $\tilde{\theta}$

• ConUCB - Method

The strategy of arm selection is

$$a_t = \arg \max_{a \in \mathcal{A}_t} \underbrace{x_{a,t}^T \boldsymbol{\theta}_t}_{\tilde{R}_{a,t}} + \underbrace{\lambda \alpha_t \|x_{a,t}\|_{M_t^{-1}} + (1 - \lambda) \tilde{\alpha}_t \|x_{a,t}^T M_t^{-1}\|_{\tilde{M}_t^{-1}}}_{C_{a,t}}$$

Exploitation

Exploration

where M_t is the function of $\boldsymbol{\theta}$ and $\tilde{\boldsymbol{\theta}}$ ¹¹¹

The core strategy to select arms and key-terms:

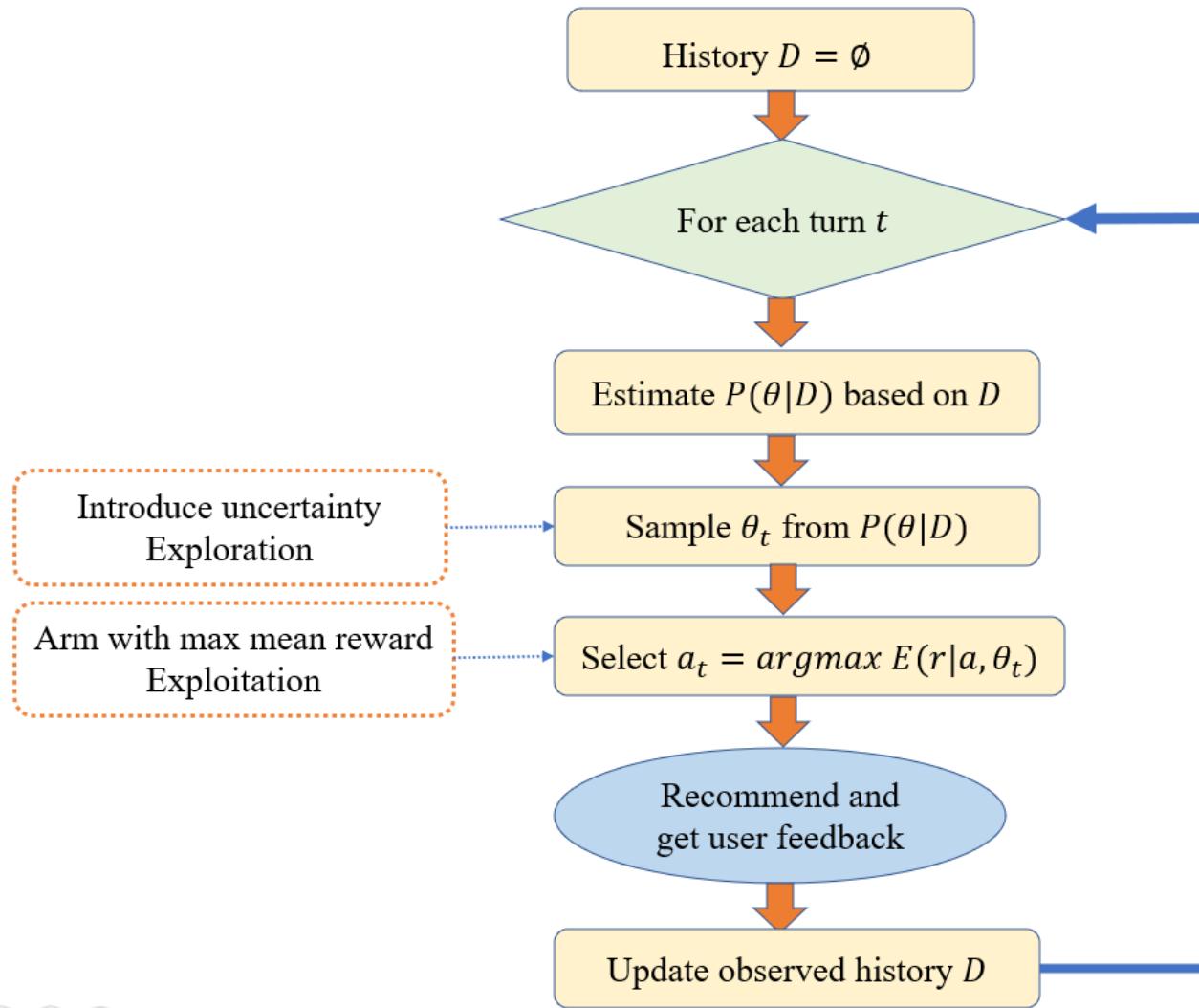
- Selecting the key-terms that maximum the reward of the corresponding items.

$$k = \arg \max_{k'} \left\| \mathbf{X}_t M_t^{-1} \tilde{M}_{t-1}^{-1} \tilde{\mathbf{x}}_{k',t} \right\|_2^2 / \left(1 + \tilde{\mathbf{x}}_{k',t}^T \tilde{M}_{t-1}^{-1} \tilde{\mathbf{x}}_{k',t} \right)$$

where $\tilde{\mathbf{x}}_{k',t} = \sum_{a \in \mathcal{A}} \frac{w_{a,k}}{\sum_{a' \in \mathcal{A}} w_{a',k}} \mathbf{x}_{a,t}$.

• Thompson Sampling

- **Bayesian bandit problem:** instead of modeling the probability of reward as a scalar, Thompson Sampling assumes the user preference comes from a distribution



•Contextual Thompson Sampling

- Assume that user preference comes from a **multidimensional Gaussian distribution**.

Arm selection strategy:

$$\hat{a} = \arg \max_a \mathbf{x}_a^T \boldsymbol{\theta}_u$$

$\boldsymbol{\theta}_u$ denotes user preference. In each turn, it is sampled from a Gaussian distribution:

$$\mathcal{N}(\boldsymbol{\mu}_u, l^2 \mathbf{B}_u^{-1})$$

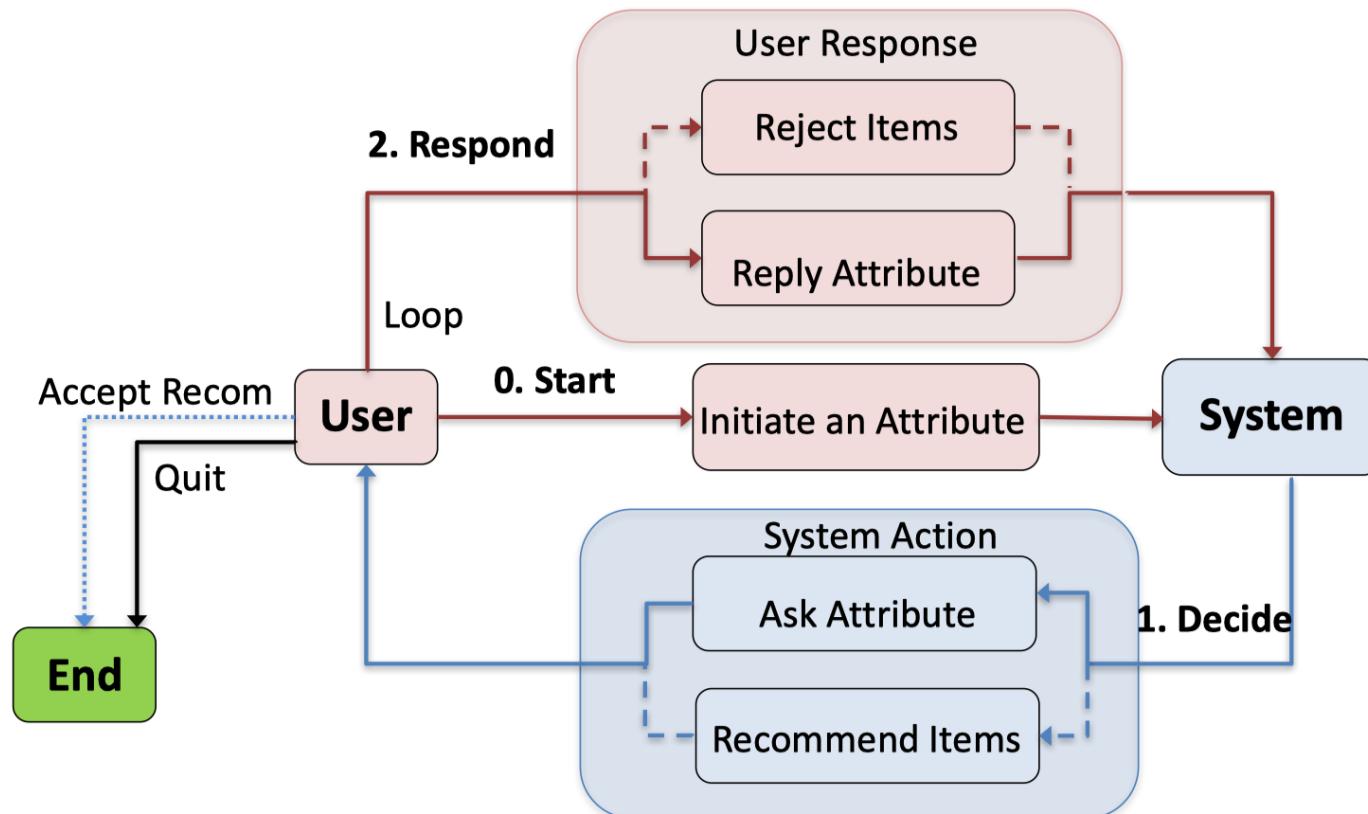


• Revisit Multi-Round Conversational Recommendation Scenario

This time, we focus on cold-start users

Objective:

Recommend desired items to user in shortest turns

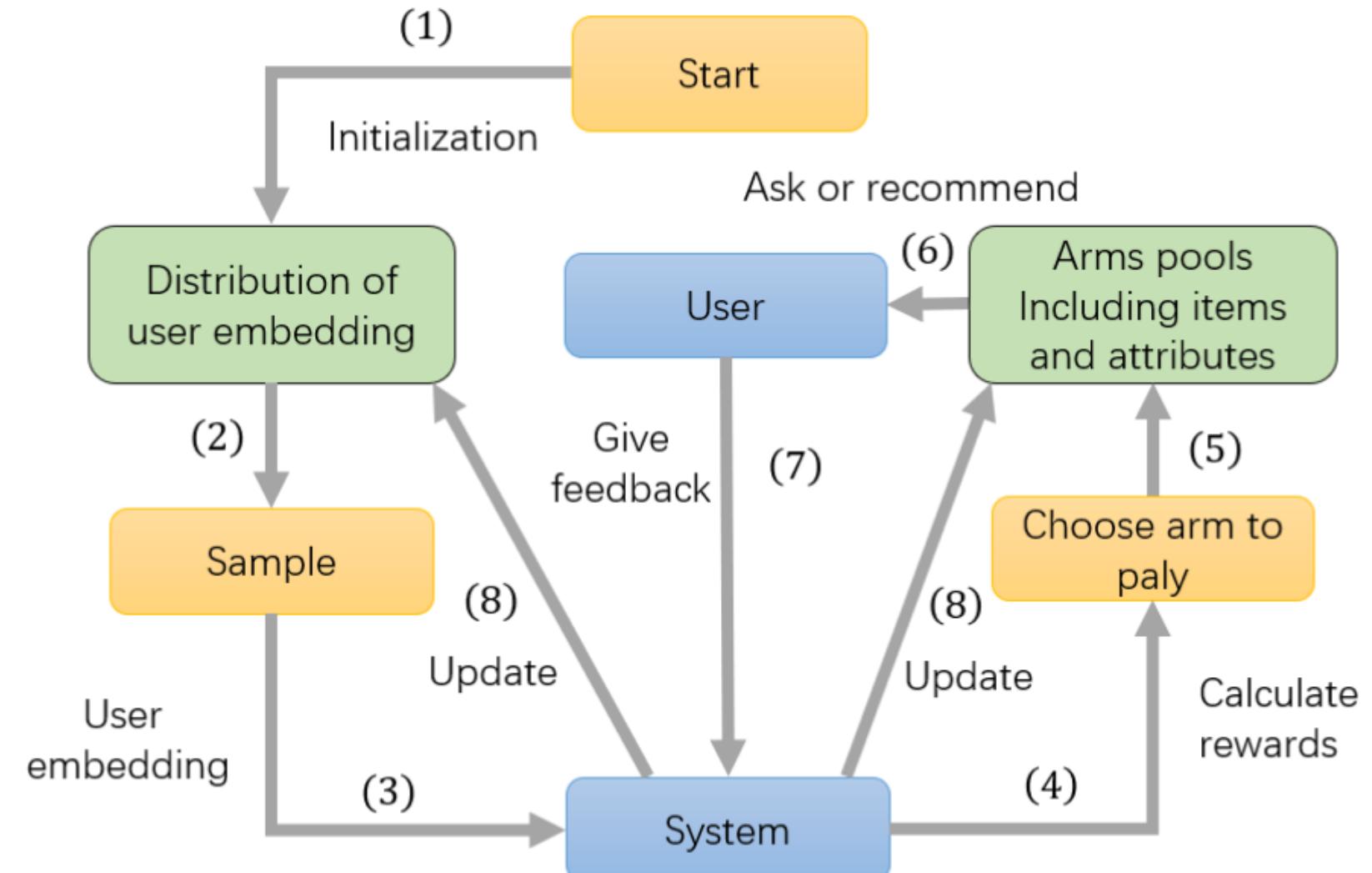


Lei et al. “Estimation–Action–Reflection: Towards Deep Interaction Between Conversational and Recommender Systems” (WSDM’20)

• ConTS (Conversational Thompson Sampling) -- Workflow

Treat items and attributes as **indiscriminate arms**.

Make theoretical customization for contextual TS to adapt to cold-start users in conversational recommendation.



• ConTS -- Method -- Arm Choosing

$$\mathbb{E}[r(a, u, \mathcal{P}_u)] = \mathbf{u}^T \mathbf{x}_a + \sum_{p_i \in \mathcal{P}_u} \mathbf{x}_a^T \mathbf{p}_i,$$

Arm Choosing: It is very simple, selecting the arm with highest reward.

Indiscriminate arms for items and attributes:

- If the arm with highest reward is attribute: system asks.
- If the arm with highest reward is item: system recommends top K items.

We addresses the strategy for recommendation issue by our indiscriminate designs of arms.

• ConTS -- Method -- Update

$$\mathbb{E}[r(a, u, \mathcal{P}_u)] = \mathbf{u}^\top \mathbf{x}_a + \sum_{p_i \in \mathcal{P}_u} \mathbf{x}_a^\top \mathbf{p}_i,$$

Update of Arm Pool: \mathcal{P}_u

- If user rejects an item / attribute: remove them from arm pool.
- If user likes an attribute: append it to the known attribute set for better estimation and narrow down the candidate item pool accordingly.

Update parameters of : $\mathcal{N}(\boldsymbol{\mu}_u, l^2 \mathbf{B}_u^{-1})$

$$\mathbf{B}_u = \mathbf{B}_u + \mathbf{x}_{a(t)} \mathbf{x}_{a(t)}^\top$$

$$r'_a = r_a - \mathbf{x}_{a(t)}^\top (\mathbf{u}_{init} + \sum_{p_i \in \mathcal{P}_u} \mathbf{p}_i)$$

$$f_u = f_u + r'_a * \mathbf{x}_{a(t)}$$

$$\boldsymbol{\mu}_u = \mathbf{B}_u^{-1} f_u$$

The known preferred attributes are used to estimate reward of arms as well as narrow down the candidate item pool.

• ConTS -- Evaluation -- User Simulator

User ID: 333, Item ID: 666

Item Name: “*Small Italy Restaurant*”
Item Attributes: [Pizza, Nightlife, Wine, Jazz]



Check, I don't want
“Small Paris”



I'd like some Italian food.

Yes!

Yes!

Rejected!

No!

Check, I don't want
“Rock Music”

Accepted!

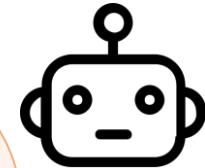
Got you, do you like some pizza?

Got you, do you like some nightlife?

Do you want “Small Paris”?

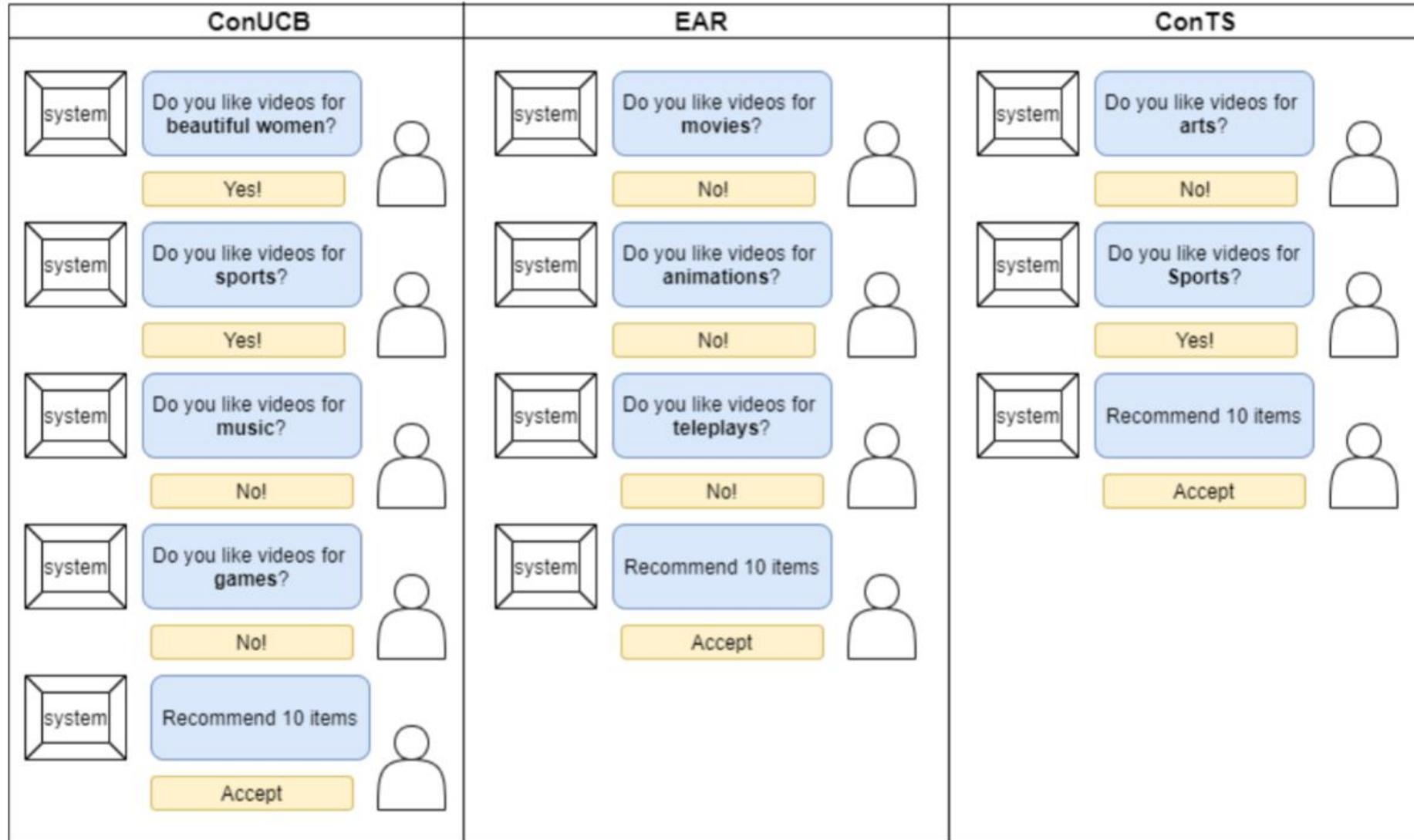
Got you, do you like some Rock Music?

Do you want “Small Italy Restaurant”?



Template-based
utterances

• ConTS -- Evaluation-- Case Study on Kuaishou



ConTS unifies items and attributes and keeps EE balance.

• VDA IRS -- Formalization

A Visual Dialog Augmented Interactive Recommender System

Yu et al. (KDD'19)

Round 1



I prefer
boots.

I prefer
blue color.

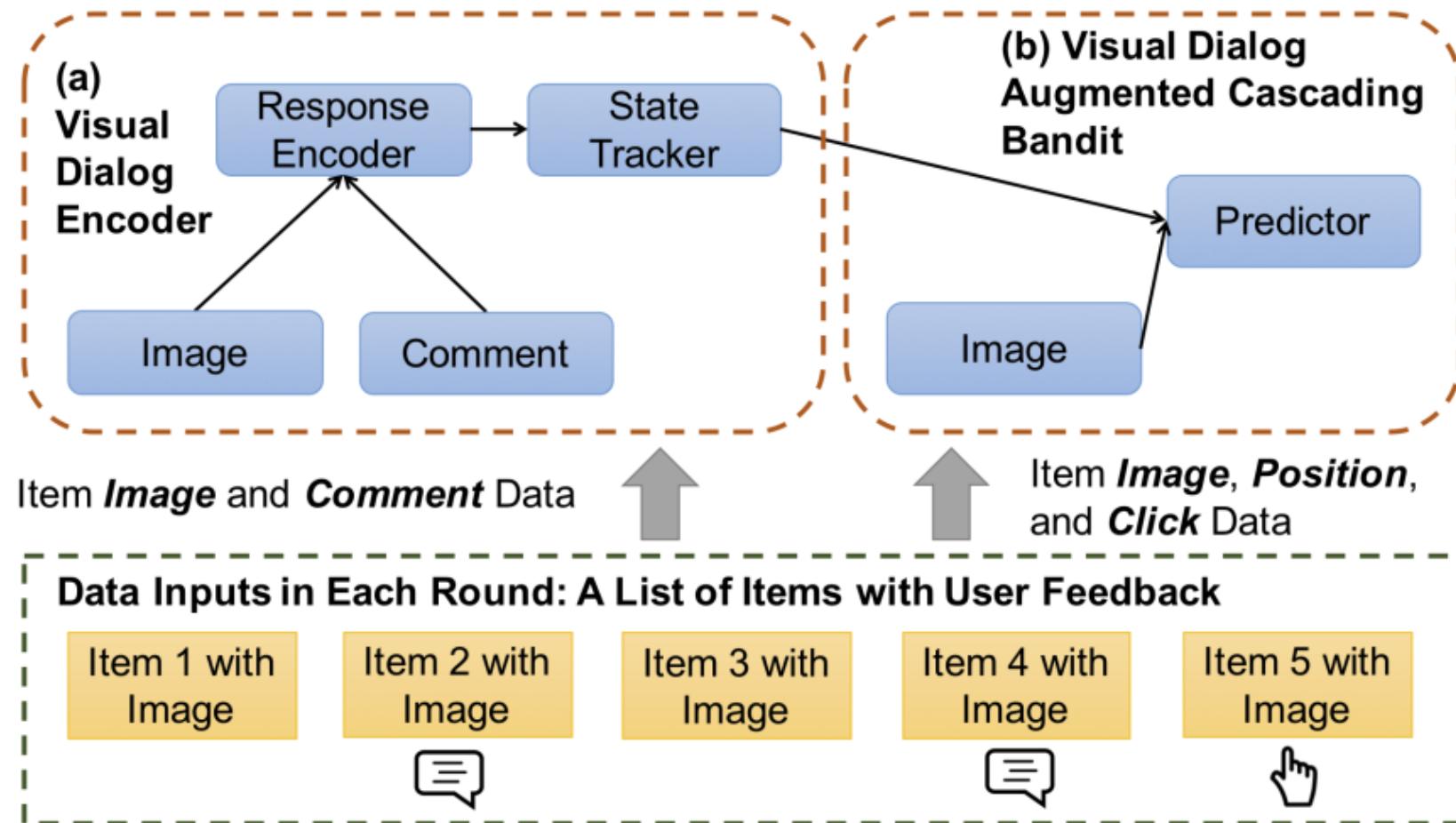
Round 2



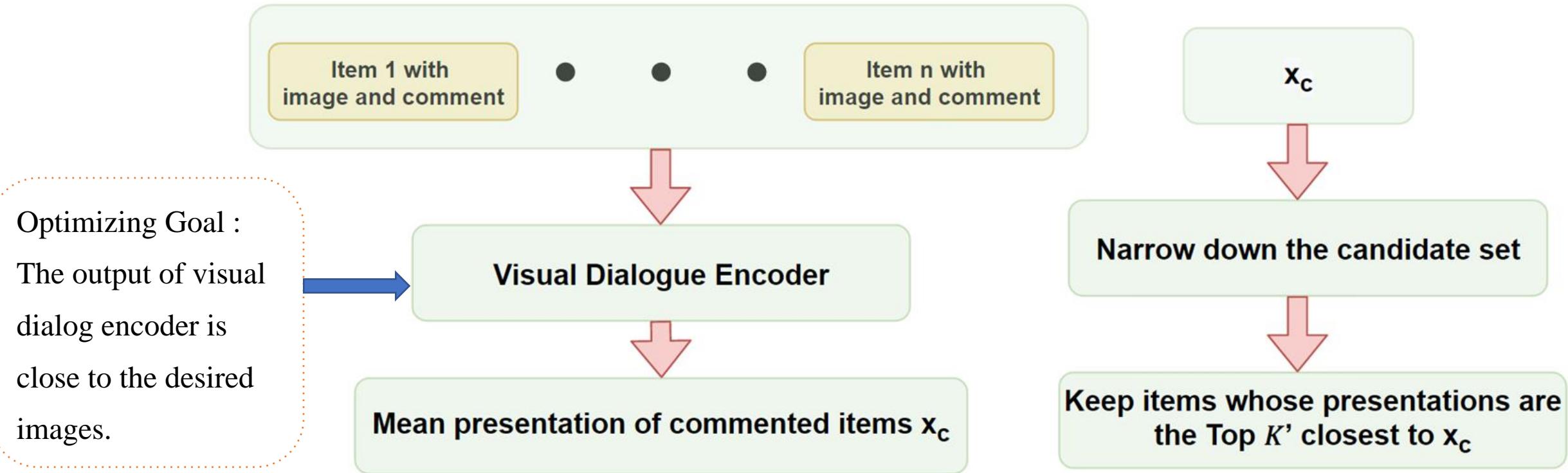
I prefer
high boots.

I prefer
high heel.

• VDA IRS -- Workflow



• VDA IRS -- Method -- Visual Dialog Encoder



The **comments** and images are encoded to help elicit the user preferences and narrow down the candidate set.

• VDA IRS --Method--Visual Dialog Augmented Cascading Bandit

forall $t = 1, \dots, n$ **do**

 Sample the parameter θ from its posterior

$$\theta_t \sim \mathcal{N}(\bar{\theta}_t, S_t)$$

forall $k = 1, \dots, K$ **do**

$$| \quad a_k^t \leftarrow \arg \max_{e \in \mathcal{L} - \{a_1^t, \dots, a_{k-1}^t\}} x_e^\top \theta_t$$

end

$$A_t \leftarrow (a_1^t, \dots, a_k^t)$$

 Observe click $C_t \in \{1, \dots, K, \infty\}$

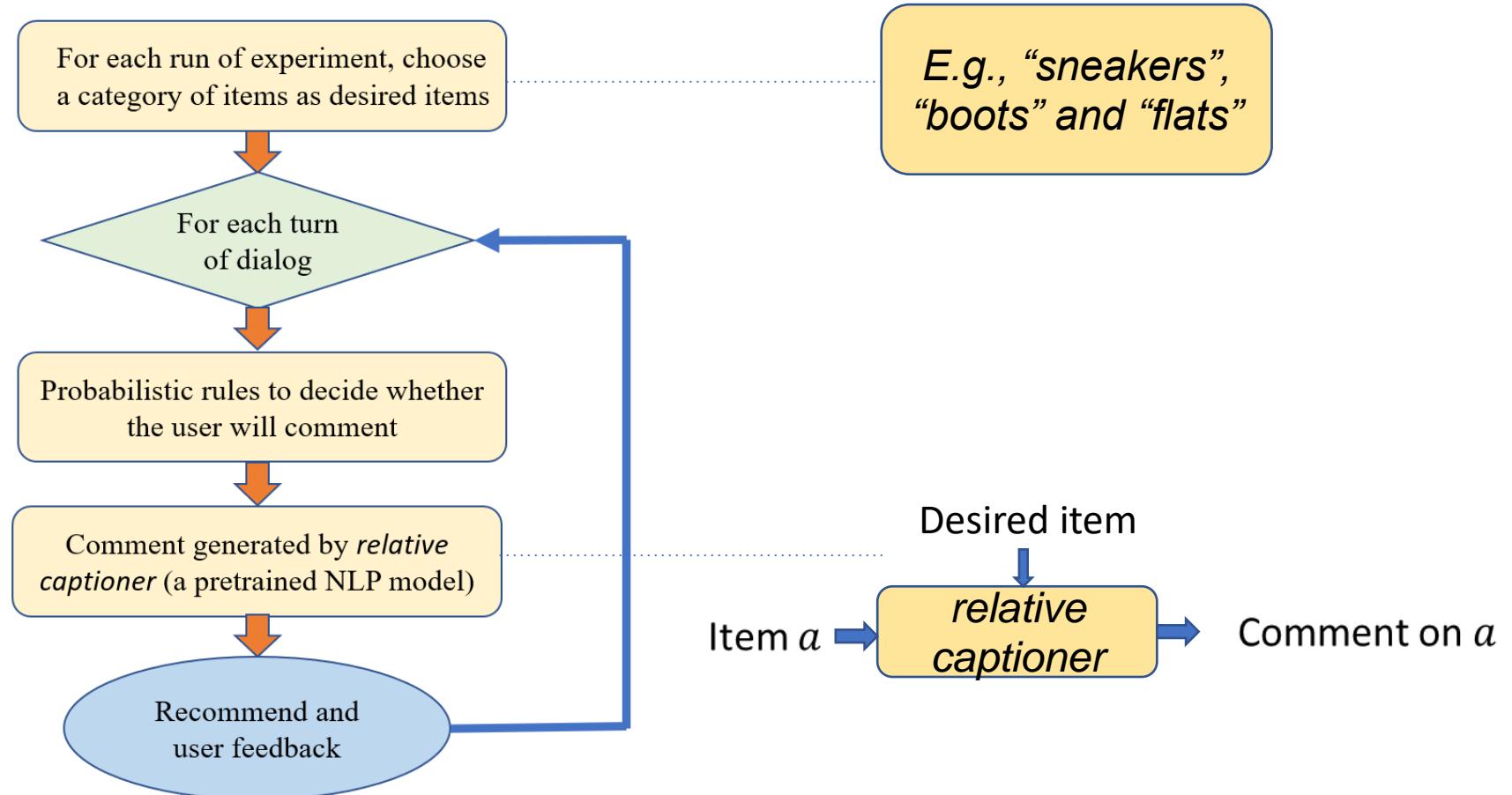
Each turn, the model will recommend a list of the Top K items.

• VDA IRS -- Evaluation

Dataset:

- A footwear dataset where 10,000 images for offline training the visual dialog encoder and 4,658 images for evaluating different interactive recommenders.

User simulator:



- Strategies in the conversational recommendation bandit (ConUCB)

Evaluation setting for real data:

- How to simulate users' ground-truth rewards on unobserved arms?

1. Use interactions of test set as known rewards $r_{a,t}$
2. Given users' feature $x_{a,t}$ on an arm a .
3. Estimate users' preferences θ using ridge regression:

$$\theta = \arg \min_{\theta} \sum_{t=1}^{|T_a|} (x_{a,t}^T \theta - r_{a,t})^2 + \|\theta\|^2$$

4. Simulate the ground-truth reward on unobserved arm and key terms of by this estimated θ .

• Tutorial Outline

- ❑ A Glimpse of Dialogue System
- ❑ Four research directions in conversational recommendation system
 - ❑ Question Driven Approaches
 - ❑ Multi-turn Conversational Recommendation Strategy
 - ❑ Dialogue Understanding and Generation
 - ❑ Exploitation-Exploration Trade-offs for Cold Users
- ❑ **Summary of Formalizations and Evaluations**

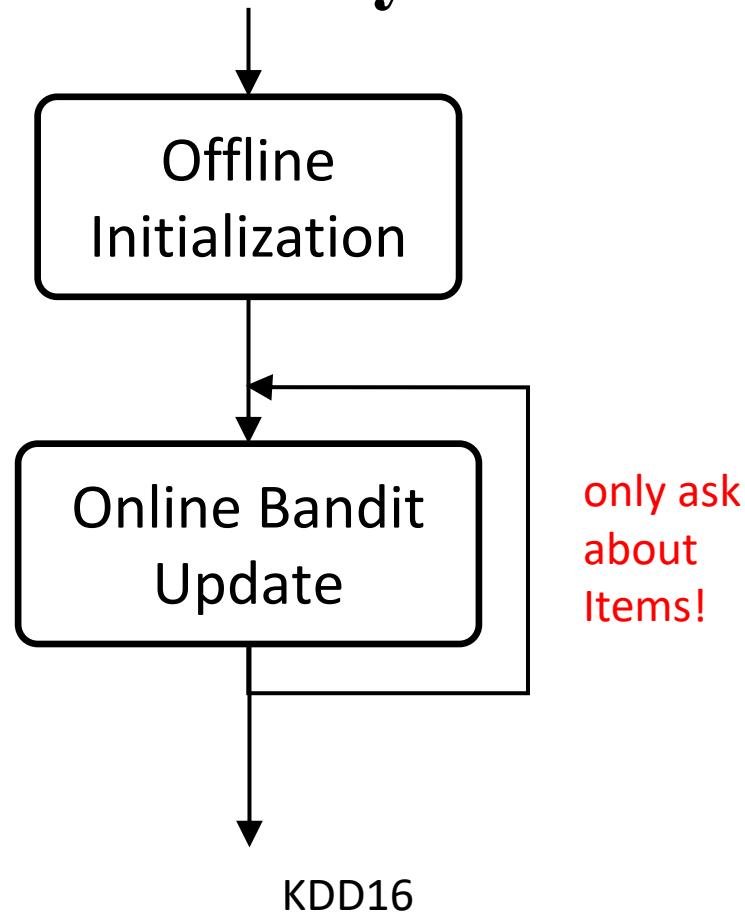


• Summary – Formalization

Mainstream settings for CRS:

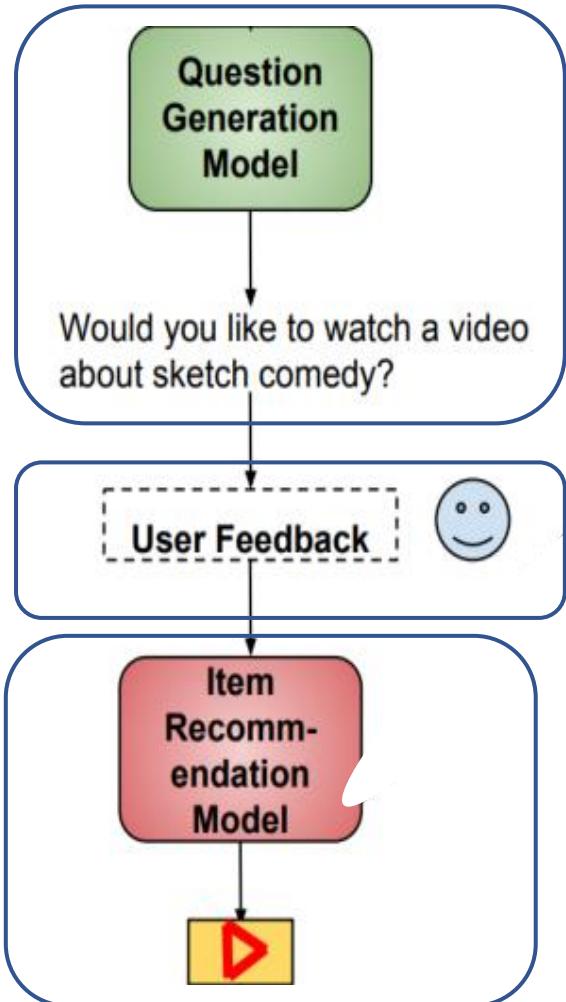
- Only consult on items.
- Ask 1 turn, recommend 1 turn.
- Ask X turn, recommend 1 turn (X is predefined).
- Ask X turn, recommend 1 turn (The system need to decide X).
- Ask X turn, recommend X turn.
- Natural Language Understanding and Generation.

• Summary – Formalization – Only Consulting on Items



- The system only consult users on their preference on items.
- Cannot leverage on the advantage of explicitly consulting on attributes.

• Summary – Formalization – Ask 1 Turn, Recommend 1 Turn



- The session will end regardless the recommendation successes or not.



- The session will continue till the recommendation successes.

Figure 1: High-level Q&R overview.

Christakopoulou et al. "Q&R: A Two-Stage Approach toward Interactive Recommendation"(KDD' 18)

Yu et al. A Visual Dialog Augmented Interactive Recommender System (KDD' 19)

• Summary – Formalization – Ask X Turns, Recommend 1 Turn



Are you seeking for a
cotton related item?



Yes!



Are you seeking for a
beach towel related
item?



No.



Are you seeking for a
bath-room towel related
item?



Yes!

The recommendation
list:



Towel A

Towel B

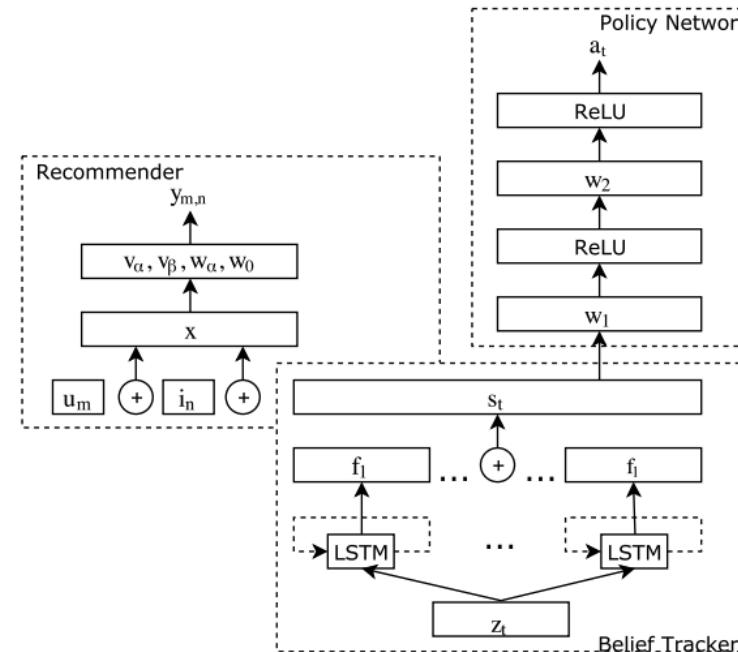
The number of questions $X = 3$
is a hyper-parameter which is
specified.

- Ask K question and then recommend one batch of items. (**X is pre-defined**)
- Do not take long-term strategy into account.



Making Recommendation
only once N_q questions

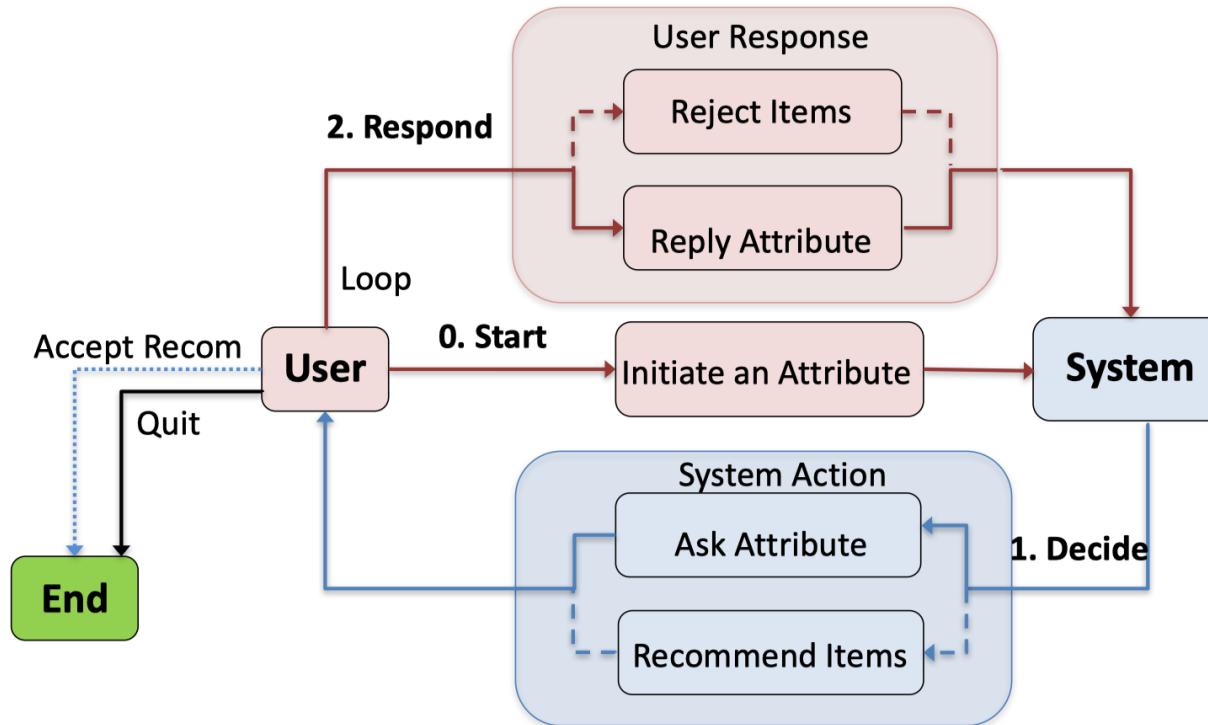
• Summary – Formalization – Ask X Turn, Recommend 1 turn



Zhang et al. Conversational Recommender System SIGIR18

- Ask X question and then recommend one batch of items. (**X is decided by model**)
- The session will end regardless the recommendation succeeds or not.
- Only consider strategy in a shallow way (e.g. after asking 3, 4 or 5 question, should I recommend?)

• Summary – Formalization – Ask X turn, Recommend X turn



Lei et al. "Estimation–Action–Reflection: Towards Deep Interaction Between Conversational and Recommender Systems" (WSDM'20)

- Ask X question and then recommend one batch of items.
- The session **will go on** even if the recommendation is not successful!

• Summary – Formalization – Natural Language Understanding and Generation

HUMAN: hello, can i help you find a movie ?

HRED: hi what kind of movies do you like

OURS: hi, how can i help you ?

SEEKER: hello ! i would like to find a suspenseful, but clean family friendly movie .

...

SEEKER: 2001 : a space odyssey might be a great option. anything else that you would suggest ?

HUMAN: you can try planet of the apes the older one is quite suspenseful and family friendly .

HRED: i haven't seen that one yet but i heard it was good. i haven't seen that one. have you seen the last house on the left ?

OURS: star wars : the force awakens is also a good one return of the jedi all good movies

SEEKER: those sound good ! i 'm going to look into those movies.

HUMAN: i hope you enjoy, have a nice one

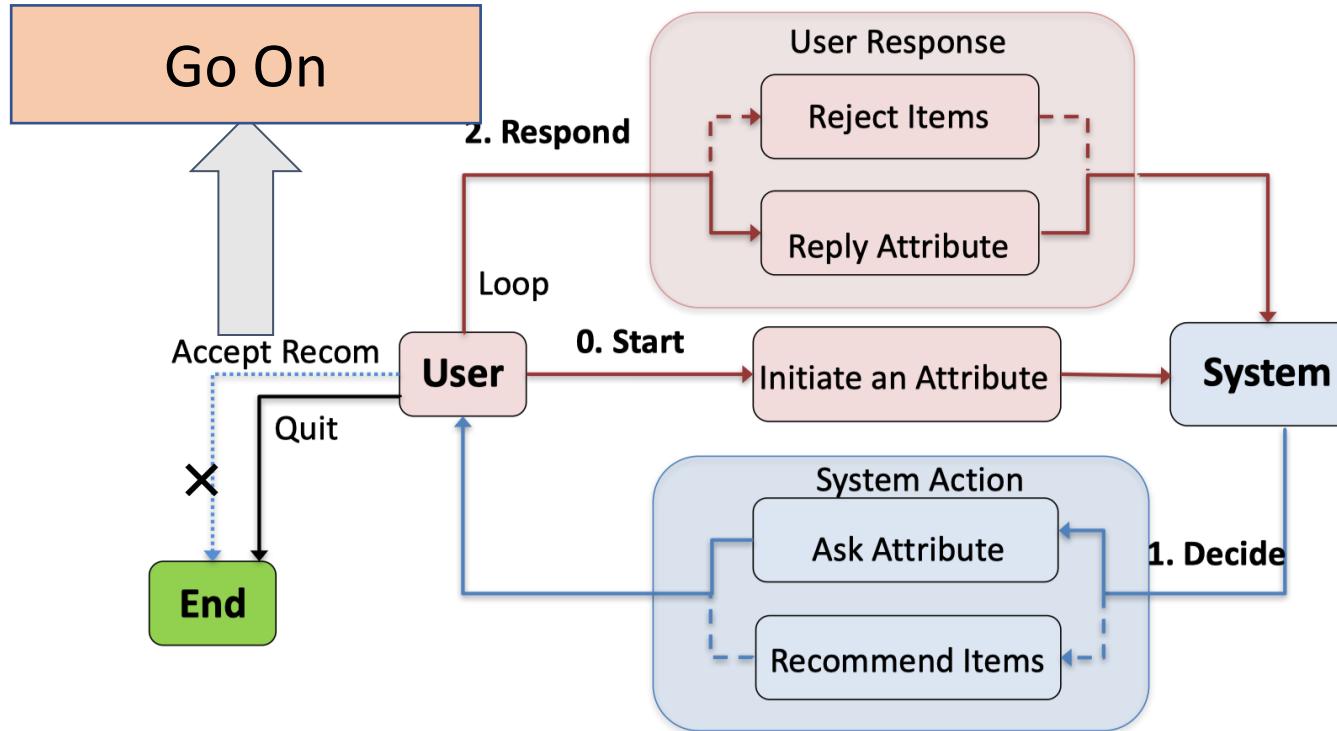
HRED: have you seen foxcatcher ? it 's about a man who has a rich guy.

OURS: i hope i was able to help you find a good movie to watch

SEEKER: thank you for your help ! have a great night ! good bye

- This is more likely to be a special type of dialogue system. More popular in NLP community.

• Summary – Formalization – Future Directions



The session will go on even if the recommendation is successful.

- Maximize Profit
- Increase the time users stay



• Summary – User Preference Simulation

Mainstream approaches to simulate user preference:

- **User click history:** EAR (WSDM20), CPR (KDD20), CRM(SIGIR18)
- **Generalize to the full datasets:** (KDD16) ConUCB (WWW20)
- **Extract from user review:** SAUR (CIKM18)
- **Corpus based:** the line of NLU/NLG works

• Summary – User preference simulation – User Click History

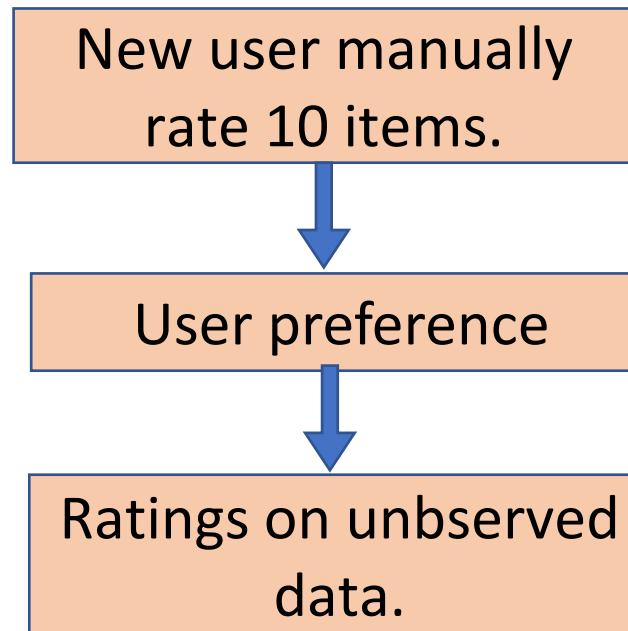
User Click History:

- Observed (user – item) pairs are used as positive samples, unobserved once as negative samples.
- During one conversation session, we sample one (user – item) pair.
 - During this session, the user will only like this item.
 - During this session, the user will only like the attributes of this item.

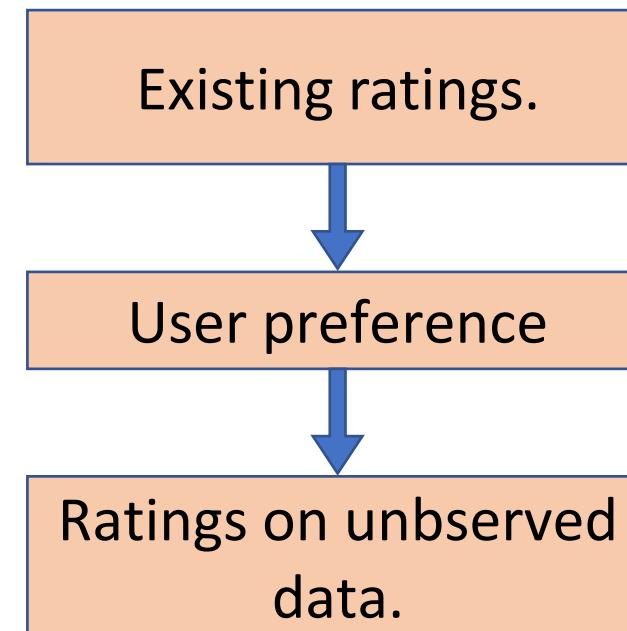


• Summary – User preference simulation – Generalize to the Whole Candidate Testing Set

- Get user's ground-truth preference score on a small amount of data.
- Infer user's preference for the full dataset.



Christakopoulou et al. “Towards Conversational Recommender Systems” (KDD’ 16)

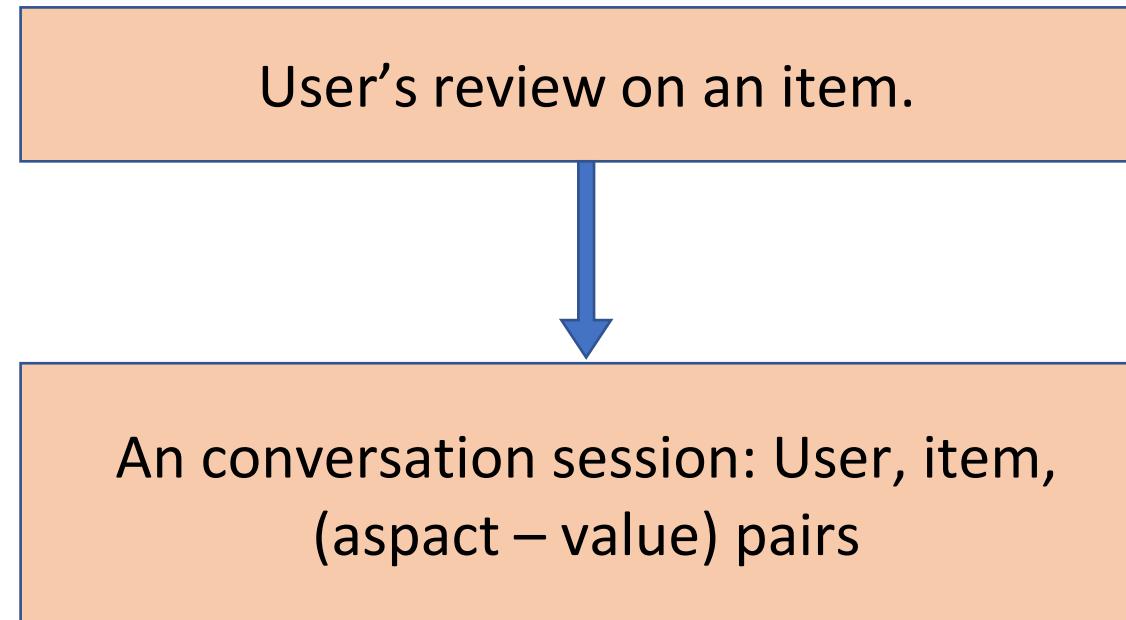


Zhang et al. “Conversational Contextual Bandit: Algorithm and Application” (WWW’ 20)

• Summary – User preference simulation – Extract from User Review

Extract from user review:

- Each review will be used to generate a conversation session.
- “Aspect – Value” pairs would be extracted from the review (e.g. “price” = “high”, ‘OS’ = “Android”).



• Summary – User preference simulation – Corpus based

HUMAN: hello, can i help you find a movie ?
HRED: hi what kind of movies do you like
OURS: hi, how can i help you ?
SEEKER: hello ! i would like to find a suspenseful, but clean family friendly movie .

...

SEEKER: ...
2001 : a space odyssey might be a great option. anything else that you would suggest ?

HUMAN: you can try planet of the apes the older one is quite suspenseful and family friendly .

i.e. corpus

User actually likes “Star Wars” and dislikes “the planet of the apes”.

Conversational recommendation through natural language.

- User’s preference is recorded “as is” in the corpus. The evaluation is actually biased on responses in the corpus (which is often generated on AMTurker).

• Discussion on Future Researches

Formalization (problem setting):

- If a user accepts the recommendation, is it possible to recommend more?
- Can we optimize other goals other than clicking? For example, maximizing profits in E-commerce; maximizing total time spending in video sharing platform ...

Evaluation (simulating user preferences):

- How to reliably simulate user preferences and action in conversational recommendation scenarios!